LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

DTIC ACCESSION NUMBER

LEVEL

INVENTORY

DOCUMENT IDENTIFICATION

DISTRIBUTION STATEMENT

ACCESSION FOR
NTIS GRA&I ☒
DTIC TRAC ☐
UNANNOUNCED ☐
JUSTIFICATION

BY
DISTRIBUTION/
AVAILABILITY CODES

DISTRIBUTION	AVAILABILITY AND/OR SPECIAL	
A-1		

DISTRIBUTION STAMP

DATE ACCESSIONED

Reproduced From
Best Available Copy

DATE RETURNED

19981223 035

DATE RECEIVED IN DTIC

REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

H A N D L E W I T H C A R E

NADC

Tech. Info!

8000748

AVIONICS DIVISION ITT

390 Washington Avenue, Nutley, N.J. 07

REPORT NUMBER D 11804

REED SOLOMON ENCODER / DECODER

COMPUTER PROGRAM PACKAGE

FEBRUARY,1976

CNI SYSTEMS ENGINEERING

Prepared Under Contract: N62269-75-C-0503

Data Item A006

for

Department of the Navy, Naval Air Development Center

by

ITT Avionics Division

500 Washington Avenue

Nutley, New Jersey 07110

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Reed Solomon Encoder/Decoder Final Engineering Report COMPUTER PROGRAM PACKAGE		5. TYPE OF REPORT & PERIOD COVERED FINAL ENGINEERING REPORT 18 JUN 75 - 9 FEB 76
		6. PERFORMING ORG. REPORT NUMBER D 11804
7. AUTHOR(s) STAFF OF: CNI Systems Engineering		8. CONTRACT OR GRANT NUMBER(s) N62269-75-C-0503
9. PERFORMING ORGANIZATION NAME AND ADDRESS ITT Avionics Division 500 Washington Avenue Nutley, New Jersey 07110		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Design, Fab & Test RSED
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Air Development Center Warminster, Pennsylvania 18974		12. REPORT DATE February 9, 1976
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Reed-Solomon
Encoder
Decoder
Round Trip Timing
Microprocessor
Galois Field
Errata

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The purpose of this program is to build a Reed-Solomon encoder/decoder capable of a 384 word/second (approx. 57.6 KBPS) throughput rate in a cost effective and practical manner.

20. ABSTRACT

ITT Avionics has built and successfully tested a RSED laboratory breadboard that was funded under contract N62269-75-C-0503 (Naval Air Development Center). A summary of the engineering tests is listed below.

Encoding Time	<150 microseconds
Round Trip Timing Detection	< 20 microseconds
Decode Time	Decode time is dependent on Errata. Refer to section 5 of report number D11801 for decode times.

COMPUTER PROGRAM PACKAGE

(MICROPROCESSOR)

This data item contains the following items:

1. Object Program Listing

2. Object Program (Note- The object program is physically contained in hardware ROMS that are part of the deliverable hardware).

3. Source Program Listing

4. Intel 3000 Computer (Microprocessor) Data Sheets

OBJECT PROGRAM LISTING

RSED OBJECT PROGRAM

MOD 1

```
000  03' FC' 03' FD' 31' FE' 03' 91'
008  00' 6C' 08' 4C' 00' 10' 00' 00'
010  E3' FC' 03' 03' 00' E3' 03' FC'
018  08' 4C' 10' 0C' 7C' 28' 03' F8'
020  FC' 03' 03' 9F' FC' 03' FF' 02'
028  10' 0C' 20' 18' 04' 73' 0F' FE'
030  7C' FC' 03' 03' FC' 02' 03' 03'
038  28' 18' 40' 30' 48' 44' 03' 00'
040  03' FC' 03' DD' FC' 02' 00' FC'
048  40' 30' 14' 60' 08' 1C' FF' FC'
050  83' 03' 03' 7F' C7' 9B' 03' 00'
058  14' 60' 28' 54' 14' 58' C3' 02'
060  FE' 00' FF' 03' FE' 03' 03' 03'
068  28' 54' 50' 3C' 4C' 70' FF' F8'
070  C3' FC' 7F' 03' BF' 02' FF' 7C'
078  50' 3C' 34' 78' 2C' 68' 03' 02'
080  01' 03' 01' 03' FF' FF' FF' FC'
088  34' 78' 68' 64' 0C' 54' 00' FC'
090  7C' 03' FC' 03' FF' 13' FF' 93'
098  68' 64' 44' 5C' 74' 64' 03' 02'
0A0  03' 02' 03' 03' 03' 33' 4F' FC'
0A8  44' 5C' 1C' 2C' 18' 24' 02' 00'
0B0  FF' 03' 7F' 02' A3' FF' 53' FC'
0B8  1C' 2C' 38' 58' 6C' 40' FF' FC'
0C0  03' FF' 07' 00' FF' FF' FC' E7'
0C8  38' 58' 70' 24' 50' 34' FF' 00'
0D0  FC' 87' 03' 03' FF' FC' 03' 02'
0D8  70' 24' 74' 48' 20' 38' 03' 00'
0E0  7C' 00' 00' 00' FC' 02' FC' FC'
0E8  74' 48' 7C' 04' 30' 60' FC' 00'
0F0  7F' 00' 03' FF' 02' 02' 00' 02'
0F8  7C' 04' 6F' 03' 5C' 3C' 02' 00'
100  7C' C7' 03' 7C' 03' 00' 13' 9B'
108  FC' FC' 01' FF' FC' 7F' FF' 00'
110  03' 03' 00' 00' 7C' 03' FC' 9B'
118  00' E3' 01' FF' 02' FC' FF' 03'
120  7C' 02' 03' BF' 00' 02' 00' 02'
128  02' 7F' 03' 03' 03' FC' FF' FF'
130  00' FF' 1F' FF' 03' 02' 03' 00'
138  00' 7F' 00' 03' 7F' E3' FC' FC'
140  FC' FF' BF' FF' 03' 03' FF' 00'
148  FC' FF' 00' 03' FF' 03' 00' 08'
150  FC' 09' 83' FF' FC' 03' 03' C0'
158  7F' 03' 7F' 03' 03' 02' FF' 03'
160  FF' FF' B7' F3' 03' FF' FF' 7C'
168  7F' 03' 03' 01' 03' 00' 03' FF'
170  03' 03' 03' 4F' 03' 03' F3' BF'
178  03' FF' FF' FF' FC' 00' E2' EF'
180  FF' FC' FF' 7F' FC' 03' FF' 03'
188  02' 03' 03' C3' 00' FF' 02' FF'
190  03' FC' 03' 03' FC' 27' 03' 03'
198  03' FC' 00' FF' FC' 83' 03' FF'
1A0  00' 00' 02' 02' FF' 03' FC' FC'
1A8  01' 7F' 03' 03' 00' 9F' 7F' 03'
1B0  FC' 00' FF' 03' FF' 03' 03' 00'
1B8  FC' 9C' 7F' E3' 03' FF' 03' FF'
1C0  FC' 00' 00' 01' FC' 04' FF' 03'
1C8  FF' 03' 03' FF' FF' 03' 02' 1F'
1D0  00' 00' 03' 03' 02' FF' FF' FF'
1D8  03' 03' 03' 03' 03' 7F' E7' 03'
1E0  FC' 03' 00' FF' FC' FC' FC' 00'
1E8  03' F4' 02' 07' 03' 7A' 7F' 5F'
1F0  03' 00' 03' 03' 03' 03' FC' 7F'
1F8  01' 00' FF' 03' FF' FF' 03' 02'
FIN
```

MOD

000	80	BF	80	BF	85	84	80	86
008	30	80	80	30	80	30	80	80
010	80	BF	80	30	30	87	80	BF
018	80	80	80	80	80	80	30	81
020	BF	85	30	86	BF	80	BF	80
028	80	30	80	80	30	80	30	BF
030	80	BF	30	80	BF	80	30	80
038	80	80	80	80	80	80	30	80
040	80	BF	80	08	BF	80	30	BF
048	80	80	80	80	80	80	BF	81
050	85	85	80	80	85	86	84	80
058	80	80	80	80	80	80	86	80
060	BF	80	BF	80	BF	30	80	87
068	80	80	80	30	80	30	BF	83
070	FF	BF	80	80	BF	30	BF	80
078	80	80	80	30	80	80	30	80
080	80	80	80	80	BF	BF	BF	BF
088	80	80	80	80	80	80	30	83
090	80	80	BF	80	BF	80	BF	86
098	80	80	80	30	80	80	30	80
0A0	80	80	80	80	30	80	80	BF
0A8	80	80	80	80	80	80	80	80
0B0	BF	80	30	80	80	BF	80	BF
0B8	80	80	30	80	80	80	BF	81
0C0	80	BF	80	80	BF	BF	BF	86
0C8	80	80	80	80	80	30	BF	80
0D0	BF	85	80	80	BF	BE	80	80
0D8	80	80	80	80	80	80	80	84
0E0	80	85	80	80	BF	80	BF	BF
0E8	80	80	80	80	80	80	BF	80
0F0	80	80	80	BF	80	80	80	80
0F8	80	80	80	80	80	30	30	00
100	80	85	80	80	30	80	30	86
108	BF	BF	80	BF	30	80	BF	80
110	80	80	30	80	84	80	BF	85
118	80	86	80	BF	84	BF	BF	80
120	80	85	80	BF	80	80	80	80
128	80	80	80	80	80	BF	BF	81
130	84	BF	80	BF	80	30	84	80
138	80	80	80	30	80	80	BF	BF
140	BF	BF	BF	BF	80	80	BF	85
148	BF	BF	80	80	BF	80	85	80
150	BF	86	BF	BF	BF	80	80	80
158	80	80	80	30	80	80	BF	80
160	BF	BF	BF	BF	80	BF	BF	80
168	80	80	80	80	80	80	30	BF
170	80	80	80	80	80	80	BF	86
178	80	BF	BF	BF	BF	80	86	BF
180	BF	BF	BF	80	BF	80	84	30
188	80	80	80	86	80	BF	80	BF
190	80	BF	80	80	BF	30	80	80
198	80	BF	80	BF	BF	BF	80	BF
1A0	80	80	84	80	BF	80	BF	BF
1A8	80	80	30	84	80	86	BF	80
1B0	BF	80	BF	84	BF	80	80	80
1B8	BF	86	80	86	87	BF	80	BF
1C0	BF	80	30	32	BF	80	BF	80
1C8	BF	30	80	BF	BF	30	80	80
1D0	80	80	85	84	80	BF	BF	BF
1D8	80	80	80	80	80	80	86	80
1E0	BF	80	80	BF	BF	BF	BF	80
1E8	80	BF	80	80	80	80	87	87
1F0	84	80	80	84	80	30	BF	80
1F8	80	85	BF	80	BF	BF	80	80

FIN

MOD 3

000	04	04	01	C8	04	CC	08	34
008	B8	B8	B8	B8	B8	B8	3C	04
010	08	08	18	C8	08	99	CC	08
018	B8	B8	B8	B8	B8	B8	74	08
020	0C	0C	CC	DC	0C	0C	04	99
028	B8	B8	B8	B8	B8	B8	0C	0C
030	10	60	D8	D4	10	54	CC	5C
038	B8	B8	B8	B8	B8	B8	10	10
040	14	14	98	02	14	04	7C	0C
048	B8	B8	B8	B8	B8	B8	2C	14
050	18	18	04	C8	18	10	30	10
058	B8	B8	B8	B8	B8	B8	18	18
060	1C	1C	D8	04	1C	14	1C	14
068	B8	B8	B8	B8	B8	B8	1C	1C
070	20	20	3C	B9	20	18	14	64
078	B8	B8	B8	B8	B8	B8	20	20
080	24	28	1C	B9	24	08	D4	1C
088	E8	B8	B8	B8	B8	B8	50	24
090	28	28	C0	C4	28	98	60	20
098	B8	B8	B8	B8	B8	B8	28	28
0A0	2C	34	B9	90	2C	2C	98	24
0A8	B8	B8	B8	B8	B8	B8	20	2C
0B0	30	CC	C4	C0	30	98	64	28
0B8	B8	B8	B8	B8	B8	B8	30	30
0C0	34	C0	D4	D8	34	20	4C	2C
0C8	B8	B8	B8	B8	B8	B8	14	34
0D0	38	30	90	C0	48	38	2C	38
0D8	B8	B8	B8	B8	B8	B8	08	38
0E0	44	38	D0	81	40	3C	40	F8
0E8	B8	B8	B8	B8	B8	B8	54	3C
0F0	48	38	CC	18	C4	C4	38	B5
0F8	B8	B8	B8	B8	B8	B8	FC	40
100	3C	D4	CC	C0	3C	34	30	3C
108	48	44	DC	F8	E4	50	04	44
110	00	40	D0	D0	D4	48	70	48
118	40	E0	DC	F8	40	40	08	74
120	4C	44	C4	6C	54	D0	68	4C
128	4C	58	D8	F8	20	44	34	4C
130	50	44	49	D0	10	B9	10	50
138	50	48	F8	F4	48	48	24	50
140	54	54	C4	D0	58	D8	24	54
148	54	28	FC	5C	4C	54	0C	54
150	58	4C	D8	14	08	1C	20	58
158	58	24	F0	B9	58	58	00	18
160	5C	5C	D8	D4	C4	6C	28	04
168	5C	54	FC	F8	50	5C	70	5C
170	60	D0	D8	D4	60	28	34	60
178	60	58	69	FC	E8	60	60	60
180	64	C0	D8	D4	64	50	3C	78
188	64	5C	F0	E4	64	64	68	64
190	28	78	10	D4	68	68	18	18
198	2C	50	E4	F8	70	68	34	68
1A0	6C	10	C4	C0	2C	1C	E0	30
1A8	30	4C	E4	F0	5C	6C	6C	10
1B0	70	64	D0	C4	4C	24	20	68
1B8	70	E0	14	DC	78	70	F0	94
1C0	34	D0	D4	C8	74	74	DC	74
1C8	DC	74	B9	E4	F4	74	FC	6C
1D0	78	60	C4	C0	78	24	6C	E0
1D8	D8	78	F0	F4	44	14	5C	F8
1E0	7C	70	D0	4C	D4	3C	44	D8
1E8	38	7C	E0	F8	38	FC	F4	7C
1F0	84	64	D4	C4	38	D0	E8	14
1F8	7C	6C	F8	F0	F4	DC	EC	48

FIN

MOD 4

```
000  20' DE' 67' 26' 60' 66' 80' 40'
008  A0' A0' A0' A0' A0' A0' 58' A0'
010  BE' 66' E0' 20' 46' 38' E0' 61'
018  A0' A0' A0' A0' A0' A0' A0' A0'
020  5E' C6' A0' 40' 66' 01' 78' 00'
028  A0' A0' A0' A0' A0' A0' A0' B3'
030  66' FE' 20' 38' 1E' 20' 20' E0'
038  A0' A0' A0' A0' A0' A0' E0' 00'
040  36' E6' 38' 0A' 66' 00' E0' 40'
048  A0' A0' A0' A0' A0' A0' 53' 00'
050  9E' A6' 38' 20' 5E' 40' 01' A0'
058  A0' A0' A0' A0' A0' A0' 60' A0'
060  E6' 06' 58' F8' 1E' E0' 00' A0'
068  A0' A0' A0' A0' A0' A0' 13' A0'
070  66' 66' 00' 00' 66' A0' 80' 60'
078  A0' A0' A0' A0' A0' A0' 18' A0'
080  5F' 86' 01' 00' DE' 80' 30' 78'
088  A0' A0' A0' A0' A0' A0' 58' B3'
090  66' 07' 7E' FE' FE' 20' 98' 40'
098  A0' A0' A0' A0' A0' A0' 13' 20'
0A0  87' 86' 06' 06' 26' A0' 00' 60'
0A8  A0' A0' A0' A0' A0' A0' 13' 20'
0B0  66' 86' A6' 06' A6' 38' A0' 53'
0B8  A0' A0' A0' A0' A0' A0' 38' 20'
0C0  1F' 66' 60' 40' 3E' E1' 40' 40'
0C8  A0' A0' A0' A0' A0' A0' B8' 40'
0D0  26' 86' 06' 5E' 9E' 66' E0' E3'
0D8  A0' A0' A0' A0' A0' A0' E0' 40'
0E0  66' 67' 06' 06' 66' 7E' 60' 18'
0E8  A0' A0' A0' A0' A0' A0' 60' 38'
0F0  E6' FE' 80' 38' 5E' 5E' 00' 00'
0F8  A0' A0' A0' A0' A0' A0' 38' 00'
100  66' 5E' E6' BE' 7E' 06' 01' 40'
108  E0' 60' A0' A0' 38' 60' A1' A0'
110  8D' 46' 06' 1A' 66' 3E' 60' 40'
118  00' 58' 78' A0' A0' 40' A1' 00'
120  7E' BF' 5E' 47' FE' 26' 40' 18'
128  80' F8' 18' E0' 59' 98' 78' A0'
130  46' 7E' 26' 06' 5F' 06' 20' 00'
138  20' E0' 40' 20' C0' B8' 60' 58'
140  DE' E6' 7E' 7E' DE' F8' E0' 60'
148  60' 81' 00' 38' 80' 80' 61' 60'
150  66' 66' 40' 00' 65' 38' 00' 00'
158  E0' 19' 98' 00' 40' 98' A1' 21'
160  FE' FE' A0' A0' C6' 18' 38' C0'
168  78' 38' 33' 40' 00' C0' 58' 98'
170  C6' 86' 80' 00' 06' 80' A0' A0'
178  80' 38' 58' 98' E0' 60' 40' 60'
180  26' 66' 98' E0' 3E' 60' 60' A0'
188  78' A0' 98' 00' 00' 13' 06' 80'
190  9F' 66' 20' E0' 66' A0' 31' B3'
198  D9' 40' E0' 40' 60' C0' C1' C0'
1A0  A6' E6' A6' 36' BF' 01' 60' 60'
1A8  61' E0' C0' A0' 40' 60' 06' 41'
1B0  46' 06' 06' A6' 06' 20' E1' 00'
1B8  60' 38' F9' 40' 3E' 38' 26' A0'
1C0  67' FA' 80' 00' E5' 40' F8' E0'
1C8  F8' A0' 00' D3' 18' D3' 20' B8'
1D0  E6' E6' E6' A6' 3F' 01' 80' 98'
1D8  60' 00' 33' D8' E0' F9' 40' A0'
1E0  E6' 7E' A6' 06' 46' 67' D8' 30'
1E8  3F' B3' 1E' 66' 67' 66' 46' 46'
1F0  36' 06' 7E' A6' 3F' FE' C0' 01'
1F8  00' 40' 80' C0' 30' 30' C0' 20'
FIN
```

MOD 5

000	01	04	09	0B	03	03	01	03
008	03	03	03	03	03	03	02	09
010	03	0B	03	00	08	05	08	0B
018	03	03	03	03	03	03	09	03
020	04	02	09	03	01	00	04	0C
028	03	03	03	03	03	03	03	01
030	0B	04	00	03	04	00	0A	03
038	03	03	03	03	03	03	03	08
040	08	0F	02	FF	0B	01	03	0B
048	03	03	03	03	03	03	04	02
050	02	03	00	0A	02	03	02	00
058	03	03	03	03	03	03	03	09
060	03	0C	05	00	05	03	00	03
068	03	03	03	03	03	03	04	03
070	0D	0B	02	0C	03	00	01	0B
078	03	03	03	03	03	03	03	09
080	02	02	03	0C	04	0B	0F	04
088	03	03	03	03	03	03	06	03
090	0B	0C	04	07	04	02	04	03
098	03	03	03	03	03	03	02	09
0A0	02	03	0C	0C	01	03	02	0B
0A8	03	03	03	03	03	03	03	08
0B0	02	02	03	08	03	04	03	04
0B8	03	03	03	03	03	03	04	02
0C0	00	02	0A	00	04	0A	0B	03
0C8	03	03	03	03	03	03	04	08
0D0	04	02	0C	02	04	05	03	03
0D8	03	03	03	03	03	03	03	02
0E0	0B	03	0C	0C	0B	02	0B	05
0E8	03	03	03	03	03	03	0B	00
0F0	07	06	00	04	02	02	09	0C
0F8	03	03	03	03	03	03	00	0C
100	01	02	0F	04	02	0C	03	03
108	0F	0B	09	0B	04	03	03	09
110	02	08	0C	0C	0B	02	0B	08
118	0C	03	01	0B	03	01	03	0C
120	05	03	02	02	06	02	08	03
128	03	04	03	03	06	04	04	03
130	03	04	03	03	00	0C	02	0C
138	00	07	00	01	0B	03	0B	04
140	04	04	03	04	06	03	00	03
148	01	01	0C	00	0F	03	03	0B
150	0B	03	03	02	0B	03	00	0C
158	07	02	05	0C	08	02	03	08
160	04	04	03	03	00	04	04	0D
168	05	00	00	02	01	03	00	05
170	00	00	00	02	02	00	03	03
178	03	04	04	05	0F	01	03	03
180	02	0B	04	02	04	00	03	03
188	00	00	02	0A	0C	05	08	0F
190	02	0B	05	03	0B	03	00	01
198	00	0B	00	0A	0B	02	00	0B
1A0	00	00	02	02	04	03	0B	0B
1A8	08	00	00	03	00	03	02	02
1B0	0B	0C	02	03	03	03	03	0C
1B8	0B	05	04	03	02	05	08	01
1C0	01	06	00	03	0F	0B	04	03
1C8	04	05	0C	04	05	00	01	03
1D0	00	08	02	03	02	04	00	04
1D8	00	03	03	00	03	04	03	09
1E0	0F	02	00	02	0B	01	04	01
1E8	02	04	06	03	09	03	03	03
1F0	03	0C	06	03	02	06	0F	0A
1F8	0C	03	0F	00	0F	0B	03	01

FIN

MOD 6

```
000  09' 00' 02' 03' 03' 02' 03' 00'
008  7F' 03' 6F' 03' 03' 01' FD' 03'
010  00' FC' 73' 01' 00' 03' 03' FC'
018  90' 03' C0' 40' 92' E3' 00' FC'
020  40' 01' 0B' 00' FC' 02' FC' FC'
028  6F' 03' 6F' FF' 6F' 03' FE' FF'
030  03' FC' 03' 03' 03' FF' FC' FC'
038  02' FF' 03' FF' FF' FC' 01' 02'
040  FF' FC' EF' EF' FC' 03' FF' FC'
048  6F' 03' 03' 03' 03' FC' FF' 00'
050  7F' FF' 80' 30' FC' FF' FF' FF'
058  82' 92' 00' C0' FF' FC' 03' 02'
060  B7' 00' 01' 03' 03' 03' C0' 03'
068  FF' 7F' 7F' 7F' 7F' FF' 03' 03'
070  9B' FC' 03' 03' 00' FF' FC' 01'
078  D6' 90' 92' 90' FC' 03' FF' FC'
080  03' 03' 03' 00' FC' 03' FF' FC'
088  FF' 7F' 6F' 6F' 03' 9B' 03' 00'
090  03' 03' 02' 03' 03' FF' 91' 00'
098  92' 92' 03' 03' 03' 00' FC' 00'
0A0  03' 03' 02' 0A' FC' 00' FC' FC'
0A8  6F' 00' 03' 93' 00' FC' 03' FC'
0B0  27' FC' 02' 01' 00' 01' 00' FC'
0B8  80' 00' 03' 00' FC' FC' 7F' 03'
0C0  4B' 02' 03' 03' FF' FF' 01' 03'
0C8  7D' FC' 03' 03' FC' FF' 00' 03'
0D0  03' 00' 00' 00' FE' 03' 00' 00'
0D8  80' 00' 03' 03' 08' 03' FC' FF'
0E0  03' FC' FF' FF' 00' 03' 00' FC'
0E8  03' 1F' FF' 03' 03' 03' 00' 03'
0F0  FF' FF' 90' 90' 03' FC' FF' 02'
0F8  FE' 03' FC' FD' FC' FF' FF' 00'
100  FC' 6F' 03' FC' FC' FC' 00' 03'
108  D5' 03' FB' FB' 00' 03' 03' FC'
110  0F' 00' FF' 03' 03' 03' FC' 03'
118  26' FE' 02' 02' 00' 03' 02' 00'
120  FC' FC' 03' 02' 00' 03' C7' 00'
128  03' 03' 03' 03' 67' 03' 7F' FF'
130  FC' 03' FC' 03' FC' FC' FC' 03'
138  2A' 03' 03' 03' FF' 67' 03' 00'
140  00' 00' 03' 03' FC' 00' FC' 03'
148  D9' FF' FF' FF' FF' AB' FC' FC'
150  03' FF' FF' 03' 03' FF' FF' 00'
158  2A' FF' 0B' 01' 02' 03' 00' 01'
160  00' 03' D3' DB' FF' 00' 03' 02'
168  E7' 03' 03' 03' 63' 00' 03' 00'
170  FC' 7F' FC' 00' EE' FC' 0B' FF'
178  27' FE' 26' 26' FF' FC' 03' FF'
180  73' FC' FF' 03' 03' 03' DB' 03'
188  D7' 03' FC' 03' 00' 43' 03' 1F'
190  FF' 03' 7E' 02' FC' 00' 2A' 00'
198  26' 03' 03' 03' 02' 00' FF' FF'
1A0  03' 03' 03' 03' 00' FF' 00' FC'
1A8  D9' DF' 03' 03' FC' FF' 03' FC'
1B0  FF' 00' 03' 00' 03' FC' FC' 00'
1B8  2A' 26' FC' 01' 43' FC' 03' 00'
1C0  03' FC' 03' 03' 02' 00' D9' D7'
1C8  D9' DF' 03' 03' 03' 03' AB' C7'
1D0  00' 03' 03' 03' 03' FC' 00' 23'
1D8  2A' 26' 2A' 23' 02' 02' 03' CB'
1E0  FC' 02' D5' D5' 03' FC' 00' D7'
1E8  E7' D7' 03' 03' 02' 00' FF' 67'
1F0  03' FC' 00' 03' FC' FF' FC' 00'
1F8  03' 26' 7C' 03' 03' FC' FC' 67'
FIN
```

MOD 7

000	86'	80'	80'	30'	80'	80'	80'	80'
008	00'	80'	7F'	80'	80'	80'	84'	80'
010	30'	BF'	87'	30'	30'	80'	80'	BF'
018	FF'	82'	FF'	FF'	FF'	80'	80'	3F'
020	80'	80'	86'	80'	BF'	80'	BF'	BF'
028	FB'	80'	00'	00'	FF'	80'	BF'	BF'
030	86'	BF'	80'	80'	30'	EF'	BF'	BF'
038	80'	81'	30'	5F'	BF'	BF'	80'	80'
040	3F'	3F'	00'	1D'	5F'	80'	BF'	FF'
048	1E'	82'	80'	30'	80'	BF'	BF'	80'
050	80'	BF'	00'	06'	3F'	BF'	84'	BF'
058	FF'	1E'	00'	16'	BF'	BF'	85'	80'
060	BF'	80'	80'	82'	80'	80'	5F'	80'
068	00'	00'	90'	9F'	80'	BF'	80'	30'
070	86'	BF'	80'	80'	80'	3F'	BF'	80'
078	DB'	FF'	EF'	FF'	BF'	80'	BF'	BF'
080	80'	80'	80'	80'	3F'	80'	BF'	BF'
088	7B'	7F'	EE'	68'	84'	86'	80'	80'
090	80'	80'	80'	80'	80'	9F'	86'	80'
098	76'	FF'	80'	80'	80'	80'	BF'	80'
0A0	87'	80'	80'	86'	BF'	80'	BF'	FF'
0A8	62'	80'	80'	86'	30'	BF'	80'	BF'
0B0	87'	BF'	80'	80'	80'	80'	80'	BF'
0B8	8D'	87'	80'	80'	BF'	BF'	80'	80'
0C0	87'	80'	80'	80'	BF'	BF'	80'	80'
0C8	64'	BF'	80'	80'	BF'	BF'	80'	80'
0D0	80'	80'	EF'	06'	BF'	80'	30'	80'
0D8	7E'	80'	80'	85'	30'	80'	BF'	BF'
0E0	80'	BF'	00'	28'	80'	80'	84'	BF'
0E8	84'	80'	BF'	80'	80'	80'	80'	80'
0F0	BF'	BF'	84'	00'	30'	BF'	BF'	80'
0F8	F2'	80'	BF'	BF'	BF'	BF'	BF'	80'
100	BF'	87'	80'	5F'	BF'	BF'	80'	80'
108	00'	80'	81'	83'	84'	80'	30'	BF'
110	86'	80'	BF'	80'	80'	80'	BF'	80'
118	FE'	BF'	82'	30'	80'	80'	80'	80'
120	BF'	BF'	80'	80'	80'	80'	85'	80'
128	80'	80'	30'	80'	81'	80'	30'	BF'
130	BF'	80'	BF'	80'	FF'	BF'	BF'	80'
138	FF'	82'	85'	80'	BF'	81'	80'	80'
140	80'	80'	80'	84'	BF'	84'	BF'	30'
148	95'	81'	BF'	BF'	BF'	81'	BF'	BF'
150	80'	BF'	BF'	80'	80'	BF'	BF'	80'
158	FF'	BF'	C0'	87'	30'	80'	80'	80'
160	80'	84'	00'	FF'	BF'	80'	80'	80'
168	00'	84'	84'	84'	83'	80'	80'	80'
170	BF'	80'	3F'	80'	BF'	BF'	86'	3F'
178	7E'	BF'	7F'	FF'	BF'	BF'	80'	BF'
180	87'	BF'	3F'	85'	30'	80'	9F'	80'
188	DF'	80'	BF'	80'	80'	80'	80'	30'
190	BF'	80'	87'	80'	BF'	80'	C6'	30'
198	DF'	30'	80'	84'	80'	80'	BF'	BF'
1A0	80'	80'	80'	34'	80'	BF'	30'	BF'
1A8	20'	00'	80'	80'	BF'	BF'	30'	BF'
1B0	BF'	80'	80'	80'	80'	BF'	BF'	85'
1B8	07'	09'	BF'	30'	80'	BF'	80'	80'
1C0	80'	3F'	80'	85'	30'	80'	9F'	86'
1C8	87'	80'	80'	30'	80'	30'	81'	85'
1D0	80'	80'	80'	84'	80'	BF'	8A'	06'
1D8	9E'	8F'	30'	83'	80'	30'	30'	85'
1E0	BF'	80'	00'	0C'	80'	BF'	85'	00'
1E8	FF'	FF'	80'	84'	30'	80'	BF'	83'
1F0	80'	BF'	80'	80'	BF'	BF'	BF'	85'
1F8	80'	00'	80'	80'	80'	BF'	BF'	81'

FIN

MOD 8

000	05'	05'	30'	C9'	05'	05'	25'	4D'
008	00'	11'	4F'	3D'	B5'	05'	05'	3D'
010	09'	09'	D1'	D9'	09'	09'	21'	01'
018	6D'	0D'	FF'	FF'	FF'	09'	FD'	E5'
020	0D'	0D'	B5'	DD'	0D'	0D'	0D'	05'
028	F9'	F9'	00'	00'	FF'	0D'	41'	2D'
030	11'	11'	DD'	E1'	11'	11'	11'	11'
038	9C'	09'	11'	E9'	31'	11'	FD'	45'
040	15'	15'	00'	CD'	15'	15'	15'	15'
048	4F'	0D'	E5'	F9'	35'	15'	41'	3D'
050	19'	19'	00'	48'	19'	19'	19'	0D'
058	FF'	4C'	00'	04'	11'	19'	2D'	45'
060	1D'	1D'	49'	C9'	1D'	1D'	1D'	25'
068	00'	00'	10'	FE'	39'	1D'	31'	15'
070	21'	DD'	19'	B9'	21'	21'	1D'	19'
078	AE'	F7'	F7'	FF'	19'	21'	19'	09'
080	49'	31'	C5'	DD'	25'	25'	2D'	2D'
088	EF'	AB'	0F'	0E'	FD'	25'	1D'	1D'
090	25'	2D'	49'	C9'	29'	2D'	35'	29'
098	4F'	CF'	F1'	79'	75'	29'	0D'	11'
0A0	2D'	C1'	C5'	B5'	2D'	2D'	31'	31'
0A8	35'	2D'	F9'	E5'	1D'	2D'	2D'	5D'
0B0	31'	41'	D5'	D9'	35'	31'	29'	C5'
0B8	31'	31'	FD'	F1'	21'	31'	21'	25'
0C0	55'	35'	C5'	D1'	3D'	35'	5D'	21'
0C8	8C'	35'	79'	25'	39'	35'	25'	55'
0D0	DD'	39'	CF'	45'	39'	39'	39'	39'
0D8	7D'	39'	29'	F1'	31'	39'	29'	2D'
0E0	35'	31'	00'	B1'	41'	3D'	3D'	3D'
0E8	F9'	45'	ED'	E1'	29'	3D'	35'	35'
0F0	39'	41'	12'	30'	C5'	41'	41'	49'
0F8	B8'	11'	F9'	BD'	15'	41'	15'	F1'
100	45'	45'	DD'	C1'	59'	45'	45'	05'
108	00'	45'	45'	45'	3D'	45'	01'	15'
110	3D'	49'	DD'	B5'	49'	49'	49'	5D'
118	57'	3D'	FC'	FC'	41'	49'	F1'	41'
120	35'	4D'	CD'	B9'	4D'	4D'	4D'	4D'
128	E5'	4D'	E5'	59'	45'	4D'	61'	F9'
130	11'	51'	C1'	DD'	51'	51'	51'	05'
138	FF'	51'	FD'	F9'	49'	51'	45'	51'
140	49'	1D'	DD'	C1'	15'	59'	55'	55'
148	51'	65'	59'	09'	4D'	55'	0D'	49'
150	59'	61'	D1'	D5'	5D'	19'	59'	4D'
158	FF'	11'	E5'	29'	51'	59'	51'	45'
160	5D'	55'	00'	FF'	45'	5D'	DD'	01'
168	00'	FD'	FD'	FD'	55'	5D'	55'	29'
170	61'	59'	C5'	C1'	61'	69'	09'	61'
178	FB'	61'	F7'	FF'	59'	21'	F1'	39'
180	25'	65'	D5'	5D'	6D'	7D'	7F'	65'
188	7F'	59'	F5'	F1'	69'	69'	59'	71'
190	69'	6D'	D5'	C1'	C5'	29'	3E'	69'
198	36'	5D'	F1'	F5'	ED'	6D'	61'	31'
1A0	6D'	6D'	C5'	D1'	65'	61'	2D'	6D'
1A8	0C'	00'	65'	31'	2D'	29'	65'	65'
1B0	71'	71'	75'	D9'	71'	79'	31'	01'
1B8	8C'	84'	F1'	F9'	0D'	71'	5D'	69'
1C0	75'	75'	D5'	5D'	75'	6D'	7D'	3F'
1C8	18'	00'	F5'	65'	39'	75'	69'	6D'
1D0	79'	79'	21'	D9'	79'	69'	D5'	00'
1D8	3F'	24'	80'	04'	61'	59'	71'	61'
1E0	7D'	5D'	00'	86'	95'	35'	2D'	00'
1E8	FF'	FF'	F1'	F5'	ED'	7D'	75'	75'
1F0	85'	DD'	C5'	91'	69'	D9'	79'	E1'
1F8	D1'	AC'	F9'	FD'	75'	F1'	79'	79'

FIN

MOD 9

000	66'	E6'	06'	FE'	87'	9E'	60'	7F'
008	FE'	E0'	EF'	00'	20'	20'	60'	00'
010	C6'	7E'	9E'	06'	06'	26'	06'	DE'
018	08'	61'	01'	01'	19'	A0'	20'	40'
020	64'	C6'	A6'	E4'	66'	C6'	A6'	66'
028	FF'	F8'	FE'	FF'	EF'	B8'	20'	98'
030	7E'	86'	C6'	E6'	9E'	3E'	66'	E6'
038	E6'	A0'	20'	38'	C1'	D8'	60'	18'
040	BE'	66'	FF'	77'	1E'	E6'	E6'	66'
048	FE'	A0'	E0'	00'	01'	40'	20'	01'
050	7E'	3E'	01'	01'	66'	DE'	C6'	C7'
058	01'	08'	01'	00'	C0'	F3'	81'	18'
060	66'	86'	E7'	26'	9E'	06'	E6'	E6'
068	F6'	FE'	FE'	FE'	41'	18'	60'	F8'
070	66'	7E'	27'	05'	E6'	FE'	67'	86'
078	90'	00'	09'	00'	E0'	E0'	60'	60'
080	67'	3E'	5E'	84'	66'	26'	FE'	66'
088	FE'	FE'	FE'	FF'	A0'	F8'	20'	00'
090	67'	5E'	C7'	BE'	9E'	1E'	A6'	E6'
098	11'	08'	78'	80'	B8'	80'	E0'	B8'
0A0	C6'	06'	E6'	66'	46'	7F'	64'	66'
0A8	EE'	20'	F8'	60'	00'	40'	E0'	40'
0B0	E6'	79'	05'	E6'	7E'	E6'	C6'	3E'
0B8	01'	40'	80'	60'	60'	F8'	78'	18'
0C0	06'	3E'	5E'	C6'	1E'	5E'	86'	E7'
0C8	F7'	58'	80'	01'	60'	38'	80'	40'
0D0	7E'	9E'	01'	01'	26'	BE'	06'	FC'
0D8	01'	60'	60'	80'	00'	C0'	40'	B8'
0E0	86'	67'	FE'	FE'	46'	5E'	7E'	E6'
0E8	A0'	80'	80'	B3'	F3'	80'	60'	C0'
0F0	BE'	26'	10'	00'	1E'	9E'	1E'	3E'
0F8	95'	21'	C0'	00'	C0'	98'	C0'	00'
100	66'	9E'	1E'	9E'	66'	66'	46'	DF'
108	DF'	80'	78'	78'	61'	40'	01'	61'
110	26'	26'	FE'	46'	86'	E6'	66'	5E'
118	10'	40'	20'	20'	18'	60'	18'	38'
120	66'	66'	66'	06'	E6'	86'	46'	A7'
128	E0'	E0'	20'	00'	18'	A0'	58'	E0'
130	67'	86'	46'	1E'	66'	3E'	1E'	DF'
138	A0'	A0'	A0'	40'	38'	18'	F8'	40'
140	06'	E7'	FC'	A6'	67'	66'	66'	3E'
148	E7'	A0'	78'	79'	18'	B8'	E1'	60'
150	A6'	9E'	3E'	E6'	5E'	3E'	DE'	7F'
158	00'	41'	20'	20'	E0'	D3'	79'	18'
160	86'	A6'	FF'	DF'	5E'	06'	C6'	07'
168	DF'	A0'	A0'	A0'	F3'	18'	B8'	80'
170	66'	6E'	64'	86'	86'	66'	86'	7E'
178	98'	58'	10'	18'	60'	F9'	40'	18'
180	9F'	66'	86'	06'	9E'	5E'	6F'	1E'
188	FF'	60'	40'	F8'	38'	A0'	C0'	20'
190	5E'	3E'	A6'	06'	66'	7E'	00'	26'
198	10'	F8'	18'	A0'	58'	00'	78'	D3'
1A0	E6'	3E'	66'	A6'	06'	86'	67'	66'
1A8	EF'	DF'	80'	01'	F9'	C1'	80'	F8'
1B0	3E'	3C'	3E'	3E'	06'	46'	67'	47'
1B8	00'	98'	40'	E0'	A0'	60'	00'	A0'
1C0	9E'	E6'	86'	06'	9E'	06'	CF'	4F'
1C8	EF'	DF'	78'	80'	81'	B8'	B8'	60'
1D0	9E'	5E'	07'	A6'	9E'	E6'	06'	00'
1D8	A0'	18'	00'	28'	F8'	D3'	A0'	98'
1E0	66'	1F'	DF'	DF'	26'	67'	67'	DF'
1E8	DF'	DF'	18'	A0'	78'	00'	D8'	F8'
1F0	86'	42'	86'	7E'	67'	1E'	42'	46'
1F8	06'	38'	58'	A0'	B8'	60'	60'	18'

FIN

MOD 10

000	03'	0F'	0C'	07'	08'	02'	09'	01'
008	6F'	03'	EF'	02'	09'	01'	03'	02'
010	08'	01'	02'	03'	0C'	01'	01'	04'
018	10'	03'	90'	80'	00'	03'	09'	0B'
020	0B'	02'	03'	00'	0B'	00'	01'	01'
028	EF'	03'	EF'	EF'	ED'	03'	0B'	05'
030	03'	04'	02'	02'	02'	05'	01'	03'
038	00'	03'	01'	05'	0B'	04'	02'	03'
040	04'	0B'	EF'	FD'	04'	00'	03'	0B'
048	6D'	0B'	03'	0C'	0C'	0B'	0B'	0C'
050	04'	05'	90'	90'	0B'	04'	02'	02'
058	10'	12'	90'	90'	0B'	05'	03'	03'
060	02'	00'	02'	03'	02'	01'	0B'	02'
068	7F'	EF'	6F'	EF'	03'	04'	02'	02'
070	03'	01'	09'	0C'	0F'	04'	0B'	0E'
078	12'	90'	92'	82'	07'	08'	00'	0B'
080	09'	06'	02'	00'	01'	01'	04'	01'
088	7F'	6F'	EF'	EF'	03'	02'	01'	0C'
090	09'	02'	08'	07'	02'	05'	03'	0F'
098	10'	92'	01'	01'	02'	01'	0F'	06'
0A0	02'	09'	08'	03'	04'	01'	0B'	01'
0A8	ED'	01'	06'	03'	0C'	01'	03'	0B'
0B0	02'	0B'	0C'	02'	01'	00'	08'	05'
0B8	90'	03'	09'	00'	01'	05'	04'	02'
0C0	03'	02'	02'	00'	04'	04'	08'	0F'
0C8	6F'	05'	01'	0C'	0B'	04'	01'	08'
0D0	02'	06'	10'	10'	04'	00'	0C'	06'
0D8	92'	01'	00'	03'	0C'	03'	0B'	04'
0E0	00'	0B'	6F'	6F'	00'	02'	03'	0F'
0E8	03'	03'	01'	03'	02'	01'	00'	00'
0F0	04'	02'	10'	02'	06'	04'	04'	02'
0F8	93'	01'	0F'	02'	0F'	04'	0B'	0C'
100	0B'	02'	07'	04'	01'	0B'	08'	00'
108	5E'	03'	03'	03'	03'	08'	0C'	0B'
110	02'	0F'	04'	08'	02'	0F'	0B'	00'
118	80'	0A'	03'	09'	06'	08'	03'	06'
120	0B'	0B'	09'	0C'	0F'	02'	02'	00'
128	03'	03'	01'	0C'	03'	05'	04'	07'
130	01'	02'	0B'	07'	01'	04'	05'	00'
138	A5'	0B'	03'	00'	04'	05'	02'	00'
140	0C'	0F'	06'	03'	0B'	03'	01'	02'
148	5B'	03'	04'	04'	04'	04'	0F'	01'
150	02'	04'	05'	00'	00'	05'	04'	06'
158	B1'	0A'	02'	6B'	03'	02'	06'	03'
160	08'	03'	7F'	7F'	04'	0C'	0E'	0F'
168	EE'	03'	03'	03'	02'	00'	02'	00'
170	0B'	01'	01'	08'	02'	0B'	02'	04'
178	AC'	04'	A8'	A0'	00'	05'	00'	04'
180	02'	0B'	02'	02'	02'	02'	5F'	02'
188	5E'	00'	0B'	02'	00'	03'	00'	03'
190	04'	02'	03'	01'	0B'	06'	A1'	08'
198	A0'	03'	03'	03'	06'	0C'	04'	04'
1A0	00'	02'	09'	03'	0C'	02'	03'	0B'
1A8	5F'	DE'	01'	0C'	05'	0B'	00'	07'
1B0	04'	07'	06'	06'	00'	02'	0B'	03'
1B8	21'	A8'	0B'	08'	03'	0B'	09'	02'
1C0	02'	0F'	02'	02'	06'	0C'	5B'	CF'
1C8	5F'	6E'	01'	01'	01'	02'	04'	03'
1D0	06'	02'	00'	03'	02'	0F'	00'	80'
1D8	B5'	A4'	A1'	81'	02'	02'	05'	04'
1E0	0B'	02'	FE'	FE'	00'	01'	03'	5E'
1E8	FE'	FF'	03'	03'	06'	0C'	04'	04'
1F0	02'	0B'	00'	06'	01'	04'	0B'	03'
1F8	0C'	AC'	04'	05'	02'	0B'	0B'	05'

FIN

SOURCE PROGRAM LISTING

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* SOURCE PROGRAM INITZ */												
	/* SET READ FLAG IADXX TO NOT READY, 0 to 176 RAM */												
	/* SET WRITE FLAG IADZY TO READY, 01 to 177 RAM */												
001	00 15 00	INI00	F4 R1	CLR	0000	00	00	1	0	JCC 01	00	0 to AC ,	READ
002	01 15 00		F1 R1	LMI	0176	00	00	1	0	JCC 02	00	176 to AC, 176 to MAR,	READ
003	02 15 00		F0 R1	ALR	7777	11	00	1	0	JCC 03	10	375 to AC,	WRITE
004	03 15 00		F4 R1	CLR	0000	00	00	1	0	JCC 04	00	0 to R0 ,	READ
005	04 15 00		F1 R1	LMI	0177	00	00	1	0	JCC 05	00	177 to MAR,	READ
006	05 15 00		F4 R1	CLR	0000	00	00	1	0	JCC 06	10	0 to AC,	WRITE
	/* SET READ FLAG IBDXX TO NOT READY, 0 to 376 RAM */												
	/* SET WRITE FLAG IBDZY TO NOT READY, 00 to 377 RAM */												
007	06 15 00		F1 R1	LMI	0376	00	00	1	0	JCC 07	00	376 to AC, 376 to MAR	READ
008	07 15 00		F4 R1	CLR	0000	00	00	1	0	JCC 08	10	0 to AC,	WRITE
009	08 15 00		F1 R1	LMI	0377	11	00	1	0	JCC 09	00	400 to AC, 377 to MAR	READ
	/* SET A AS CURRENT DATA BLOCK, 0 to R9 */												
010	09 15 00		F4 R1	CLR	0000	00	00	1	0	JCC 10	10	0 to R9,	WRITE
	/* MOVE TABLE CCTPC FROM ROM AND WRITE INTO RAM AS TABLE TCTPC */												
	/* MOVE TABLE CPTCC FROM ROM AND WRITE INTO RAM AS TABLE TPTCC */												

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* INITIALIZE ROM AND RAM ADDRESSES */												
011	10 15 00		F4 R1	CLR	0000	00	00	1	0	JCC 11	00	0 to R1	READ
012	11 15 00		F1 R1	LMI	0177	00	00	1	0	JCC 12	00	177 to R1,177 to MAR	READ
013	12 15 00		F4 R1	CLR	0000	00	00	1	0	JCC 13	00	0 to R2	READ
014	13 15 00		F1 R1	LMI	0400	00	00	1	0	JCC 14	00	400 to R2, 400 to MAR	READ
	/* LOAD NEXT ROM ADDRESS, LOAD OPTION SELECTS NEXT INSTRUCTION FROM TABLE */												
015	14 15 00		F0 R1	ILR	0000	11	00	1	0	JCC 15	00	ROM Address to AC	READ
016	15 15 00	INI01	F6 R1	NOP	0000	00	00	0	0	JCC 16	00		READ
017	16 15 00		F4 R1	CLR	0000	00	00	1	0	JCC 17	00	0 to AC, JCC NOT EXEC.	READ
	/* INSTRUCTION SELECTED FROM TABLE IS EXECUTED NEXT */												
	/* LOAD NEXT WORD FROM TABLE TO ACCUMULATOR. JUMP TO INI02												
	/* WRITE TABLE WORD IN NEXT RAM ADDRESS */												
018	00 14 00	INI02	F1 R1	LMI	0000	11	00	1	0	JCC 15	00	R2 to MAR,1+R2 to R2	READ
019	15 14 00		F0 R1	ILR	0000	11	00	1	0	JCR 15	10	1+R1 to R1 and AC Jump to INI01	WRITE
	/* LAST JUMP BACK FROM TABLE IS TO INI03 */												
	/* CLEAR ACCUMULATOR and EXIT TO INOUT AT IOC00 */												
020	01 14 00	INI03	F4 R1	CLR	0000	00	00	1	0	JCC 29	11	0 to AC, Remove Lock out	

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* TABLE CPTCC */												
	/* EACH INSTRUCTION IN TABLE CPTCC IS EXECUTED AFTER THE LOAD IN INI01 */												
021	00 08 00		F1 R1	LMI	0000	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
022	01 08 00		F1 R1	LMI	0002	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
023	02 08 00		F1 R1	LMI	0004	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
024	03 08 00		F1 R1	LMI	0010	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
025	04 08 00		F1 R1	LMI	0020	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
026	05 08 00		F1 R1	LMI	0005	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
027	06 08 00		F1 R1	LMI	0012	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
028	07 08 00		F1 R1	LMI	0024	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
029	08 08 00		F1 R1	LMI	0015	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
030	09 08 00		F1 R1	LMI	0032	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
031	10 08 00		F1 R1	LMI	0021	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
032	11 08 00		F1 R1	LMI	0007	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
033	12 08 00		F1 R1	LMI	0016	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
034	13 08 00		F1 R1	LMI	0034	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
035	14 08 00		F1 R1	LMI	0035	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
036	15 08 00		F1 R1	LMI	0037	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
037	00 09 00		F1 R1	LMI	0033	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
038	01 09 00		F1 R1	LMI	0023	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
039	02 09 00		F1 R1	LMI	0003	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
040	03 09 00		F1 R1	LMI	0006	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
041	04 09 00		F1 R1	LMI	0014	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
042	05 09 00		F1 R1	LMI	0030	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
043	06 09 00		F1 R1	LMI	0025	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
044	07 09 00		F1 R1	LMI	0017	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
045	08 09 00		F1 R1	LMI	0036	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
046	09 09 00		F1 R1	LMI	0031	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
047	10 09 00		F1 R1	LMI	0027	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
048	11 09 00		F1 R1	LMI	0013	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
049	12 09 00		F1 R1	LMI	0026	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
050	13 09 00		F1 R1	LMI	0011	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
051	14 09 00		F1 R1	LMI	0022	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
052	15 09 00		F1 R1	LMI	0001	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
053	00 10 00		F1 R1	LMI	0002	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
054	01 10 00		F1 R1	LMI	0004	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
055	02 10 00		F1 R1	LMI	0010	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
056	03 10 00		F1 R1	LMI	0020	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
057	04 10 00		F1 R1	LMI	0005	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
058	05 10 00		F1 R1	LMI	0012	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
059	06 10 00		F1 R1	LMI	0024	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
060	07 10 00		F1 R1	LMI	0015	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
061	08 10 00		F1 R1	LMI	0032	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
062	09 10 00		F1 R1	LMI	0021	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
063	10 10 00		F1 R1	LMI	0007	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
064	11 10 00		F1 R1	LMI	0016	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
065	12 10 00		F1 R1	LMI	0034	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
066	13 10 00		F1 R1	LMI	0035	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
067	14 10 00		F1 R1	LMI	0037	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
068	15 10 00		F1 R1	LMI	0033	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
069	00 11 00		F1 R1	LMI	0023	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
070	01 11 00		F1 R1	LMI	0003	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
071	02 11 00		F1 R1	LMI	0006	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
072	03 11 00		F1 R1	LMI	0014	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
073	04 11 00		F1 R1	LMI	0030	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
074	05 11 00		F1 R1	LMI	0025	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
075	06 11 00		F1 R1	LMI	0017	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
076	07 11 00		F1 R1	LMI	0036	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
077	08 11 00		F1 R1	LMI	0031	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
078	09 11 00		F1 R1	LMI	0027	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
079	10 11 00		F1 R1	LMI	0013	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
080	11 11 00		F1 R1	LMI	0026	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
081	12 11 00		F1 R1	LMI	0011	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
082	13 11 00		F1 R1	LMI	0022	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
083	14 11 00		F1 R1	LMI	0001	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
084	15 11 00		F1 R1	LMI	0002	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ

/* TABLE CCTPC */

/* EACH INSTRUCTION IN TABLE CCTPC IS EXECUTED AFTER THE LOAD IN INI01 */

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
085	00 12 00		F1 R1	LMI	0000	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
086	01 12 00		F1 R1	LMI	0037	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
087	02 12 00		F1 R1	LMI	0001	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
088	03 12 00		F1 R1	LMI	0022	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
089	04 12 00		F1 R1	LMI	0002	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
090	05 12 00		F1 R1	LMI	0005	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
091	06 12 00		F1 R1	LMI	0023	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
092	07 12 00		F1 R1	LMI	0013	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
093	08 12 00		F1 R1	LMI	0003	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
094	09 12 00		F1 R1	LMI	0035	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
095	10 12 00		F1 R1	LMI	0006	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
096	11 12 00		F1 R1	LMI	0033	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
097	12 12 00		F1 R1	LMI	0024	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
098	13 12 00		F1 R1	LMI	0010	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
099	14 12 00		F1 R1	LMI	0014	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
100	15 12 00		F1 R1	LMI	0027	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
101	00 13 00		F1 R1	LMI	0004	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
102	01 13 00		F1 R1	LMI	0012	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
103	02 13 00		F1 R1	LMI	0036	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
104	03 13 00		F1 R1	LMI	0021	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
105	04 13 00		F1 R1	LMI	0007	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
106	05 13 00		F1 R1	LMI	0026	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
107	06 13 00		F1 R1	LMI	0034	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
108	07 13 00		F1 R1	LMI	0032	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
109	08 13 00		F1 R1	LMI	0025	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
110	09 13 00		F1 R1	LMI	0031	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
111	10 13 00		F1 R1	LMI	0011	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
112	11 13 00		F1 R1	LMI	0020	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
113	12 13 00		F1 R1	LMI	0015	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
114	13 13 00		F1 R1	LMI	0016	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
115	14 13 00		F1 R1	LMI	0030	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ
116	15 13 00		F1 R1	LMI	0017	00	00	1	0	JZR 14	00	K to AC, K to MAR	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* SOURCE PROGRAM INOUT */														
	/* RESTORE R9 FROM SCBIX (477 RAM) */														
	/* CHECK SCBIX = ZERO FOR BLOCK A, ≠ ZERO FOR BLOCK B */														
01	00	14	01	IØC01	F1 R2	LMM	0477	00	00	1	0	JCC 01	11	477Ø to MAR	
02	01	14	01		F4 R1	CLR	0000	00	00	1	0	JCR 15	00	0 to R9	READ
03	01	15	01		F5 R2	LTM	7777	00	00	1	0	JCR 09	00	O_V M to CO M to T	READ
04	01	09	01		F1 R2	LMM	0200	00	00	1	0	JFL 03	11	200Ø to AC	
	/* BLOCK B IS CURRENT BLOCK M ≠ O */														
05	03	11	01		F2 R1	SDR	7777	11	00	1	0	JCR 10	11	200Ø to R9	
	/* BLOCK A IS CURRENT BLOCK M = 0 */														
06	03	10	01	IØC03	F0 R1	ILR	0000	00	00	1	0	JCC 04	11	R9 to AC	
	/* JUMP TO RPV02 IN RPVAL */														
07	16	14	01	IOC03	F6 R0	NOP	0000	00	00	1	0	JFL 16	11	O to R0	
	/* CHECK IF CURRENT DATA BLOCK IS BLOCK A OR BLOCK B */														
	/* IF CURRENT DATA BLOCK IS BLOCK A, R9 = 0 */														
	/* WRITE 177Ø IN RAM ADDRESS 176Ø TO SET READ DATA FLAG */														
	/* SET CURRENT DATA BLOCK TO B BLOCK R9 = 200Ø */														

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
08	16	10	01	IOC04	F1 R2	LMM	0176	11	00	1	0	JCC 17	11	176Ø to MAR	
														177Ø to AC	
09	17	10	01		F1 R1	LMI	0200	00	00	1	0	JCR 15	10	R9 + 200Ø to R9	WRITE
												P 00			

/* IF CURRENT DATA BLOCK IS BLOCK B, R9 ≠ 0 */

/* WRITE 377Ø IN RAM ADDRESS 376Ø TO SET DATA READ FLAG */

/* SET CURRENT DATA BLOCK TO A DATA BLOCK R9 = 0 */

10	16	11	01	IOC05	F1 R2	LMM	0376	11	00	1	0	JCC 17	11	376Ø to MAR
														377Ø to AC
11	17	11	01		F4 R1	CLR	0000	00	00	1	0	JCR 15	10	O to R9
												P 00		

/* INITIALIZE FOR EPVAL */

/* WRITE ZERO IN SIRPS = 671Ø RAM */

/* WRITE ZERO IN SIRCS = 670Ø RAM */

12	29	15	00		F4 R1	CLR	0000	00	00	1	0	JCR 14	11	0 to AC
13	29	14	00	IØC00	F1 R2	LMM	0671	00	00	1	0	JCC 23	11	671Ø to MAR
														671Ø to T

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
14	23	14	00		F1 R2	LMM	0670	00	00	1	0	JCC 24	10	670Ø to MAR 670Ø to T	WRITE
/* INITIALIZATION FOR EPGEN */															
/* WRITE COMMAND IS FOR INITIALIZATION OF EPVAL */															
15	24	14	00		F4 R1	CLR(R0)	0000	00	11	1	0	JCC 26	10	0 to R0	WRITE
16	26	14	00		F1 R1	LMI(R0)	7737	00	11	1	0	JCC 27	11	7737Ø to MAR 7737Ø to R0	
17	27	14	00		F4 R1	CLR(R1)	0000	00	11	1	0	JCR 12	11	0 to R1	
18	27	12	00		F1 R1	LMI(R1)	0700	11	11	1	0	JCC 30	11	701Ø to R1 700Ø to MAR	
19	30	12	00		F4 R2	CLA(AC)	0000	00	11	1	0	JFL 30	11	0 to AC	
20	30	10	00		F3 R1	INR(R0)	0000	11	11	1	0	JCR 08	10	1 + R0 to R0	WRITE
21	30	08	00		F1 R1	LMI(R1)	0000	11	11	1	0	JFL 30	11	1 + R1 to R1	
22	30	11	00		F1 R2	LDM(AC)	0001	00	11	1	0	JCR 14	11	1 to AC	
23	30	14	00		F1 R2	LMM(T)	0737	00	11	1	0	JCR 13	11	737Ø to MAR	
24	30	13	00		F1 R1	LMI(A)	0036	00	11	1	0	JCR 15	10	AC + 36 to AC	WRITE
25	30	15	00		F1 R2	LMM(T)	0733	00	11	1	0	JCC 31	11	733Ø $_V$ 0 to MAR 733Ø to T	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* LAST INSTRUCTION IN INITIALIZATION SEQUENCE */														
26	31	15	00		F0 R1	ILR	0000	00	00	1	0	JCC 18	10	R9 to AC	WRITE
27	17	15	00	IOC05	F6 R1	NOP	0000	00	00	1	0	JCC 29	11	0 to R0	
	/* READ CURRENT DATA BLOCK. WRITE CONTROL FLAG IADZY/IBDZY */														
	/* STORE ADDRESS OF WRITE CONTROL FLAG IN R2 */														
28	18	15	00	IOC09	F1 R1	LMI	0177	00	00	1	0	JCC 19	11	R9 + 177Ø to AC	
29	19	15	00		F2 R1	SDR	7777	11	00	1	0	JCC 20	00	AC to R2	READ
30	20	15	00		F5 R2	LTM	0002	00	00	1	0	JCC 21	00	$0_V M^2$ to CO M^2 to AC	READ
	/* CHECK IF CURRENT DATA BLOCK IS READY FOR PROCESSING, I.E., IADZY/IBDZY = 11 */														
31	21	15	00		F4 R1	CLR	0000	00	00	1	0	JFL 22	11	0 to R1	
	/* DATA BLOCK NOT READY */														
	/* SET DELAY COUNTER TO ZERO */														
32	22	10	00	IØC06	F0 R1	ILR	0000	11	00	1	0	JCR 15	11	1 + R1 to R1 to AC	
33	22	15	00		F2 R1	SDR	7777	11	00	1	0	JCC 23	11	AC to T	
34	23	15	00	IOC07	F1 R2	LMM	7773	00	00	1	0	JCC 24	11	7773Ø to AC	
35	24	15	00		F7 R1	XNR	7777	00	00	1	0	JCC 25	11	T $\oplus$ AC to T	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
36	25	15	00		F5 R3	TZA	7777	00	00	1	0	JCC 26	11	0_VT to C0	
	/* CHECK IF DELAY-COUNTER HAS REACHED ITS LIMIT, I.E., SDLAY = 4 */														
37	26	15	00		F1 R1	LMI	0000	00	00	1	0	JFL 20	11	R2 to MAR	
	/* LIMIT NOT REACHED */														
	/* INCREMENT DELAY COUNTER BY 1 AND RETEST */														
38	20	11	00	IOC08	F0 R1	ILR	0000	11	00	1	0	JCC 23	11	1 + R1 to R1 to AC	
39	23	11	00		F2 R1	SDR	7777	11	00	1	0	JCR 15	11	AC to T	
	/* LIMIT REACHED */														
40	20	10	00		F6 R1	NOP	0000	00	00	1	0	JCR 15	00		READ
	/* DATA BLOCK READY */														
	/* SET NO WRITE REQUEST FLAG */														
41	22	11	00	IOC10	F1 R1	LMI	0000	00	00	1	0	JCR 14	11	R2 to MAR	
42	22	14	00		F0 R1	ILR	0000	11	00	1	0	JCC 28	11	R2 + 1 to R2 to AC	
	/* SET DATA QUALITY WORD TO DECODE FAILURE */														
43	28	14	00		F0 R1	ILR	0000	00	00	1	0	JCR 15	10	R9 to AC	WRITE
44	28	15	00		F1 R1	LMI	0007	11	00	1	0	JCC 27	11	7_VR9 to MAR	
45	27	15	00		F0 R1	ALR	7777	00	00	1	0	JZR 05	10	20 or 220 to AC	WRITE

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION

/* EXIT TO APGEN AT APGOO */

/* WRITE IS IN FIRST INSTRUCTION IN APGEN */

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* SOURCE PROGRAM APGEN */												
	/* SET SCBIX BASED ON R9 */												
1	00 05 00	APG00	F1 R2	LDM	0477	00	11	1	0	JCR 03	10	MAR=SCBIX	WRITE
2	00 03 00		F5 R2	TZR	7777	00	11	1	0	JCR 02	11		
3	00 02 00		F4 R2	CLA	0000	00	11	1	0	JFL 00	11	JFL (APG 19, APG20) (AC=0)	
4	00 03 01	APG20	F3 R3	INA	0000	11	11	1	0	JCR 02	11	AC=1	
5	00 02 01	APG19	F6 R1	NOP	0000	00	11	1	0	JZR 00	10	SET SCBIX	WRITE
	/* POINT TO NUMBER OF ERASURES BASED ON R9 */												
6	00 00 00		F0 R1	ILR	0000	00	00	1	0	JCC 01	11	AC=R9	
7	01 00 00		F1 R1	LMI	0070	11	11	1	0	JCC 02	11	MAR = # OF ERASURES AC = FIRST ERASURE	
8	02 00 00		F2 R1	SDR	7777	11	11	1	0	JCC 03	00	R2 = FIRST ERASURE	READ
	/* FETCH NUMBER OF ERASURES */												
9	03 00 00		F5 R2	LTM	0037	00	11	1	0	JCC 04	00	AC = # OF ERASURES	READ
10	04 00 00		F4 R1	CLR	0000	00	11	1	0	JCC 05	11		
11	05 00 00		F1 R1	LMI	0540	11	11	1	0	JCC 06	11	MAR = TAPOL R4 = TAPOL + 1	
	/* WRITE NUMBER OF ERASURES IN TAPOL */												
	/* CHECK THAT NUMBER OF ERASURES IS BETWEEN 1 AND 16 */												
12	06 00 00		F1 R3	DCA	7777	00	11	1	0	JCC 07	10		WRITE
13	07 00 00		F6 R2	LMF	7760	00	11	1	0	JCC 08	11		

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
14	08 00 00		F1 R1	LMI	0000	11	11	1	0	JFL 09	11	JFL APG04, APG05,MAR='FIRST ERASURE"	
	/* COMPUTE ERASURE POLYNOMIAL - USE R3 AS COUNTER OF NUMBER OF ERASURES */												
15	09 02 00	APG04	F2 R1	SDR	7777	11	11	1	0	JCR 00	00	R3 = # OF ERASURES-1	READ
16	09 00 00		F5 R2	LTM	0037	00	11	1	0	JCC 10	00	AC = FIRST ERASURE	READ
	/* PLACE ERASURE IN TAPOL IF NON ZERO - OTHERWISE, REPLACE BY 37_8 IN TAPOL */												
17	10 00 00		F1 R1	LMI	0000	00	11	1	0	JFL 11	11	JFL APG 02 APG 03	
18	11 02 00	APG02	F1 R1	DSM	7777	00	11	1	0	JCR 01	11		
19	11 01 00		F1 R1	LMI	0000	00	11	1	0	JCR 03	11		
	/* INITIALIZE INNER-LOOP COUNTER TO ZERO IN R0 */												
20	11 03 00	APG03	F4 R1	CLR	0000	00	11	1	0	JCR 00	10		WRITE
	/* DECREMENT COUNT OF NUMBER OF ERASURES TO TEST FOR END OF OUTER LOOP */												
21	11 00 00		F1 R1	DSM	7777	00	11	1	0	JCC 12	11		
	*/ INCREMENT INNER-LOOP COUNTER AND STORE IN R1 */												
22	12 00 00	APG11	F0 R1	ILR	0000	11	11	1	0	JFL 13	11	JFL APG06 APG07	
23	13 02 00	APG06	F6 R1	NOP	0000	00	11	1	0	JZR 04	11	JUMP (MSYNG)	
	/* LOAD MAR WITH ADDRESS OF NEXT ERASURE AND INCREMENT ERASURE POINTER (R(2)) */												
24	13 03 00	APG07	F1 R1	LMI	0000	11	11	1	0	JCR 00	11		
25	13 00 00.		F2 R1	SDR	7777	00	11	1	0	JCC 14	00		READ

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* FETCH NEXT ERASURE AND LOAD MAR WITH ADDRESS OF BEGINNING OF OUTPUT AREA */												
26	14 00 00		F5 R2	LTM	0037	00	11	1	0	JCC 17	00		READ
27	17 00 00		F1 R1	LMI	0000	00	11	1	0	JFL 16	11	JFL (APG14, APG15)	
	/* IF ERASURE POSITION WAS ZERO, REPLACE WITH 37_8 */												
28	16 02 00	APG14	F7 R3	CMA	0000	00	11	1	0	JCR 03	11		
	/* SAVE ERASURE POSITION (POWER) IN R5 */												
29	16 03 00	APG15	F2 R1	SDR	0037	11	11	1	0	JCR 16	00		READ
	/* ADD CURRENT ERASURE TO MOST SIGNIFICANT COEFFICIENT, FOLD AROUND CARRIES, */												
	/* SAVE POWER FORM IN R6 */												
	/* FETCH CODE OF MOST SIGNIFICANT COEF. WITHOUT CURRENT ERASURE AND SAVE IN R7 */												
	/* THEN RETURN POWER FORM FROM R6 TO ACCUMULAROR */												
30	16 00 00		F0 R2	AMA	0037	00	11	1	0	JCC 15	00		READ
31	15 00 00		F3 R3	AIA	0037	00	11	1	0	JCC 18	11		
32	18 00 00		F2 R2	CSA	0037	11	11	1	0	JCC 19	00		READ
33	19 00 00		F1 R2	LMM	0400	00	11	1	0	JCC 20	00	K = TPTCC	READ
34	20 00 00		F2 R1	SDR	7777	11	11	1	0	JCC 21	00		READ
35	21 00 00		F5 R2	LTM	7777	00	11	1	0	JCC 22	00		READ
36	22 00 00		F2 R1	SDR	7777	11	11	1	0	JCC 23	11		
37	23 00 00		F0 R1	ILR	0000	00	11	1	0	JCC 24	11		

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* DECREMENT INNER LOOP COUNTER */												
	/* LOAD MAR WITH ADDRESS OF CURRENT OUTPUT LOCATION */												
	/* TEST WHETHER INNER LOOP IS COMPLETED */												
38	24 00 00	APG10	F1 R1	DSM	7777	00	11	1	0	JCC 25	11		
39	25 00 00		F1 R1	LMI	0000	11	11	1	0	JFL 26	11	JFL APG08, APG09	
	/* INNER LOOP - OVERWRITE CURRENT LOCATION WITH BETA-E TIMES PREVIOUS LOCATION PLUS CURRENT LOCATION (BETA-E X SIGMA-(I-1) + SIGMA-I) */												
	/* POINT TO NEXT LOCATION, TEST IT FOR ZERO, OTHERWISE MULTIPLY IT BY BETA-E FETCHED FROM R5*/												
40	26 03 00	APG09	F1 R1	LMI	0000	00	11	1	0	JCR 00	10		WRITE
41	26 00 00		F0 R1	ILR	0000	00	11	1	0	JCC 27	00		READ
42	27 00 00		F5 R2	LTM	7777	00	11	1	0	JCC 28	00		READ
43	28 00 00		F0 R2	AMA	7777	00	11	1	0	JFL 29	00	JFL APG16,APG17	READ
	/* IF NEXT LOCATION IS ZERO, STORE ZERO IN R7 AND CONVERT PREVIOUS LOCATION TO CODE */												
44	29 02 00	APG16	F1 R1	LMI	0400	00	11	1	0	JCR 01	11	K=TCTPC	
45	29 01 00		F4 R1	CLR	0000	00	11	1	0	JCC 24	00	JUMP APG18	READ
	/* CONVERT BETA-E X NEXT LOCATION TO CODE AND XOR TO R7 = PREVIOUS LOCATION */												
46	29 03 00	APG17	F1 R1	LMI	0400	00	11	1	0	JCR 00	11	K=TPTCC	
47	29 00 00		F0 R1	ILR	0000	00	11	1	0	JCC 30	00		READ
48	30 00 00		F7 R3	XNI	7777	00	11	1	0	JCC 31	00		READ

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* FETCH NEXT LOCATION IN CODE FORM AND STORE IN R7 */												
	/* CONVERT BETA-E X SIGMA - (I-1) + SIGMA-I TO POWER FORM */												
49	31 00 00		F1 R1	LMI	0400	00	11	1	0	JZR 01	11	K=TPTCC	
50	00 01 00		F2 R1	SDR	7777	11	11	1	0	JCC 01	00		READ
51	01 01 00		F5 R2	LTM	7777	00	11	1	0	JCC 02	00		READ
52	02 01 00		F1 R1	LMI	0400	00	11	1	0	JCC 03	11	K=TCTPC	
53	03 01 00		F2 R1	SDR	7777	11	11	1	0	JCC 24	00		READ
54	24 01 00	APG18	F5 R2	LTM	7777	00	11	1	0	JCR 00	00	JUMP APG10	READ
55	26 02 00	APG08	F1 R1	LMI	0400	00	11	1	0	JCR 01	10	K=TPTCC	WRITE
56	26 01 00		F0 R1	ILR	0000	00	11	1	0	JCC 04	00		READ
57	04 01 00		F7 R3	XNI	7777	00	11	1	0	JCC 05	00		READ
58	05 01 00		F1 R1	LMI	0400	00	11	1	0	JCC 06	11	K=TCTPC	
59	06 01 00		F6 R1	NOP	0000	00	11	1	0	JCC 07	00		READ
60	07 01 00		F5 R2	LTM	7777	00	11	1	0	JCC 08	00		READ
61	08 01 00		F1 R1	LMI	0000	00	11	1	0	JCC 10	11		
62	10 01 00		F4 R1	CLR	0000	00	11	1	0	JCC 13	10		WRITE
63	13 01 00		F1 R1	LMI	0540	00	11	1	0	JCC 12	11	K=TAPOL± 1	
64	12 01 00		F1 R1	DSM	7777	00	11	1	0	JCR 00	11		
65	09 03 00	APG05	F3 R3	INA	0000	11	11	1	0	JCR 01	11		

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION
66	09 01 00		F6 R1	NOP	0000	00	11	1	0	JFL 10	11	JFL APG12, APG13
67	10 02 00	APG12	F6 R1	NOP	0000	00	11	1	0	JZR 14	11	JMP FAILO
68	10 03 00	APG13	F6 R1	NOP	0000	00	11	1	0	JZR 04	11	JMP MSYNG

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* SOURCE PROGRAM MSYNG */														
001	00	04	00	MSY00	F1 R2	LMM	0540	00	00	1	0	JCC 01	11		
002	01	04	00		F4 R1	CLR	0000	00	11	1	0	JCC 02	00		READ
003	02	04	00		F0 R2	AMA	0017	00	11	1	0	JCC 03	00		READ
004	03	04	00		F2 R1	SDR	7777	11	11	1	0	JCC 04	00		READ
005	04	04	00		F5 R2	LTM	7777	00	11	1	0	JCC 05	00		READ
006	05	04	00		F1 R1	LMI	0561	11	11	1	0	JCC 06	11		
007	06	04	00		F2 R1	SDR	7777	11	11	1	0	JCC 07	10		WRITE
008	07	04	00		F1 R2	LDM	7757	00	11	1	0	JCC 08	11		
009	08	04	00		F2 R1	SDR	7777	11	11	1	0	JCC 09	11		
010	09	04	00		F2 R1	SDR	7777	11	11	1	0	JCC 10	11		
011	10	04	00		F0 R1	ILR	0000	00	11	1	0	JCC 11	11		
012	11	04	00		F1 R1	LMI	0050	11	11	1	0	JCC 12	11	1st Syndrome	
013	12	04	00		F2 R1	SDR	7777	11	11	1	0	JCC 13	11		
014	13	04	00		F2 R1	SDR	7777	00	11	1	0	JCC 18	11		
015	18	04	00	MSY01	F3 R1	INR	0000	11	11	1	0	JCC 21	00		READ
016	21	04	00		F5 R2	LTM	7777	00	10	1	0	JFL 18	00	JMP(MSY02, MSY05)	READ
017	18	02	00	MSY02	F1 R1	LMI	0000	11	11	1	0	JCR 01	11		
018	18	01	00		F1 R1	LMI	0000	11	11	1	0	JCF 17	10	JMP(MSY03, MSY04)	WRITE

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
019	17	02	00	MSY03	F6 R1	NOP	0000	00	11	1	0	JCR 04	00		READ
020	17	03	00	MSY04	F6 R1	NOP	0000	11	01	1	0	JCR 04	00		READ
021	17	04	00		F1 R2	LMM	0400	00	11	1	0	JCC 18	00		READ
022	17	05	00		F1 R1	LMI	0000	11	11	1	0	JCC 18	11		
023	18	05	00		F1 R1	LMI	0000	11	11	1	0	JCC 15	11		
024	18	03	00	MSY05	F1 R1	LMI	7757	00	11	1	0	JZF 19	11	JMP(MSY06, MSY07)	
025	19	02	00	MSY06	F1 R1	LMI	0007	00	11	1	0	JCC 18 PL 01	11	JMP MSY17	
026	18	02	01	MSY17	F4 R2	CLA	0000	00	11	1	0	JCR 03 P 01	11		
027	18	03	01		F6 R1	NOP	0000	00	11	1	0	JZR 14 P 01	10	JMP IOCOI	WRITE
028	19	03	00	MSY07	F1 R1	DSM	7777	00	11	1	0	JCR 04	11		
029	19	04	00		F0 R1	ILR	0000	11	11	1	0	JFL 20	11	JMP(MSY13, MSY08)	
030	20	03	00	MSY08	F2 R1	SDR	7777	11	11	1	0	JCR 04	11		
031	20	04	00		F3 R1	INR	0000	11	11	1	0	JCC 22	11		
032	22	04	00		F0 R1	ILR	0000	00	11	1	0	JCR 01	11		
033	22	01	00		F2 R1	SDR	7777	11	11	1	0	JCC 23	11		
034	23	01	00		F0 R1	ILR	0000	00	11	1	0	JCR 04	11	Address of S1	
035	23	04	00		F1 R1	LMI	0000	00	11	1	0	JCC 24	11		

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
036	24	04	00		F2 R1	SDR	7777	11	11	1	0	JCC 25	00		READ
037	25	04	00		F5 R2	LTM	7777	00	11	1	0	JCC 26	00		READ
038	26	04	00		F2 R1	SDR	7777	11	11	1	0	JFL 27	11	JMP(MSY12,MSY09)	
039	27	03	00	MSY09	F1 R1	LMI	0400	00	11	1	0	JCR 01	11		
040	27	01	00		F6 R0	NOP	0000	00	11	1	0	JCC 25	00		READ
041	25	01	00		F5 R2	LTM	7777	00	11	1	0	JCC 30	00		READ
042	30	01	00		F1 R1	LMI	0000	11	11	1	0	JCC 28	11		
043	28	01	00		F3 R1	INR	0000	11	01	1	0	JCR 04	00		READ
044	28	04	00		F7 R3	XNI	7777	00	11	1	0	JCC 29	00		READ
045	29	04	00		F1 R1	LMI	0000	11	11	1	0	JZF 30	10	JMP(MSY10,MSY12)	WRITE
046	30	02	00	MSY10	F0 R1	ILR	0000	00	11	1	0	JCR 04	00		READ
047	30	04	00		F5 R2	LTM	7777	00	11	1	0	JCR 05	00		READ
048	30	05	00		F0 R2	AMA	7777	00	11	1	0	JFL 31	00	JMP(MSY11,MSY09)	READ
049	31	02	00	MSY11	F3 R1	INR	7777	11	11	1	0	JCR 04	11		
050	31	04	00		F1 R1	LMI	0000	11	11	1	0	JFL 30	11	JMP(MSY10,MSY12)	
051	30	03	00	MSY12	F1 R1	DSM	7777	00	11	1	0	JCC 19	11	JMP MSY07	
052	20	02	00	MSY13	F1 R2	LDM	7757	11	11	1	0	JCR 01	11		
053	20	01	00		F2 R1	SDR	7777	00	11	1	0	JCC 21	11		

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
054	21	01	00		F1 R2	LDM	0602	00	11	1	0	JCC 19	11		
055	19	01	00		F1 R1	SDR	7777	11	11	1	0	JCC 17	11		
056	17	01	00		F4 R1	CLR	0000	00	11	1	0	JCC 16	11		
057	16	01	00		F1 R1	LMI	0561	11	11	1	0	JCR 05	11		
058	16	05	00		F6 R1	NOP	0000	00	11	1	0	JCC 13	00		READ
059	13	05	00		F5 R2	LTM	7677	00	11	1	0	JCC 14	00		READ
060	14	05	00		F1 R1	LMI	0000	11	11	1	0	JCC 15	10		WRITE
061	15	05	00		F1 R1	LMI	0000	11	11	1	0	JCR 01	10		WRITE
062	15	01	00	MSY14	F3 R1	INR	0000	11	11	1	0	JCC 14	00		READ
063	14	01	00		F1 R2	LMM	0500	00	11	1	0	JFL 14	00	JFL(MSY15,MSY16)	READ
064	14	02	00	MSY15	F6 R1	NOP	0000	00	11	1	0	JCR 04	00		READ
065	14	04	00		F5 R2	LTM	7777	00	11	1	0	JCC 16	00		READ
066	16	04	00		F1 R1	LMI	0000	11	11	1	0	JCC 15	11		
067	15	04	00		F1 R1	LMI	0000	11	11	1	0	JCR 01	10	JMP(MSY14)	WRITE
068	14	03	00	MSY16	F6 R1	NOP	0000	00	11	1	0	JZR 05	11	JMP(EPGEN)	
069	27	02	00		F1 R1	DSM	7777	00	11	1	0	JCR 04	11		
070	27	04	00		F1 R1	DSM	7777	00	11	1	0	JCC 19	11		
071	31	03	00		F1 R1	LMI	0400	00	11	1	0	JCR 01	11		
072	31	01	00		F6 R1	NOP	0000	00	11	1	0	JCC 25	00		READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* SOURCE PROGRAM EPGEN */														
	/* SET POINTER TO T' ARRAY */														
	/* SET COUNTER (R3) TO -(16-s) */														
001	00	00	01	EPG00	F1 R2	LDM(AC)	0602	00	11	1	0	JCC 01	11	K = TMSYP	
002	01	00	01		F4 R1	CLR(R6)	0000	00	11	1	0	JCC 02	00	0 to R6	READ
003	02	00	01		F5 R2	LTM(AC)	0020	00	10	1	0	JCC 03	00	s = 16	READ
														CO to C FLAG	
004	03	00	01		F1 R2	LMM(AC)	0602	11	11	1	0	JCC 04	00	ADDRESS OF T1	READ
														MOD SYND.	
														K = TMSYP	
005	04	00	01		F2 R1	SDR(R5)	7777	11	11	1	0	JCC 05	11	AC to R5	
006	05	00	01		F2 R1	SDR(R3)	0037	11	11	1	0	JCC 06	11	s + 3	
007	06	00	01		F1 R1	LMI(R3)	7755	00	11	1	0	JCC 07	11	(R3 = - (16-s))	
	/* TEST R9 AND SET SCBIX ACCORDINGLY */														
008	07	00	01		F1 R2	LDM(AC)	0646	00	11	1	0	JCC 08	11	K = TEVAL = 646 to MAR to AC	
009	08	00	01		F4 R2	CLA(AC)	0000	00	11	1	0	JCF 10	11	JUMP ON C FLAG	
														s = 16 TEST	
														JCF (EPG03, EPG04)	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
010	10	03	01	EPG03	F1 R2	LDM(AC)	0602	00	11	1	0	JZR 13	10	TRPOL= TMSYP	WRITE
														602 to MAR to AC	
														JMP(RPVAL)	
011	10	02	01	EPG04	F4 R1	CLR(R7)	0000	00	11	1	0	JCR 01	10	0 to R7	WRITE
012	10	01	01		F4 R1	CLR(R8)	0000	00	11	1	0	JCR 00	11	0 to R8	
	/* SET ADDRESS POINTERS */														
013	10	00	01		F1 R1	LMI(R6)	0700	00	11	1	0	JCC 11	11	K = TEPAA	
														700 to R6 + K to R6	
014	11	00	01		F1 R1	LMI(R7)	0711	00	11	1	0	JCC 12	11	K = TEPBB	
														711 to R7 + K to R7	
015	12	00	01		F1 R1	LMI(R8)	0722	00	11	1	0	JCC 21	11	K = TEPCC	
														722 to R8 + K to R8	
	/* FIRST D(U) = D(O) = T(1) */														
016	21	00	01		F1 R1	LMI(R5)	0000	00	11	1	0	JCC 22	11	R5 to MAR	
017	22	00	01		F4 R1	CLR(R4)	0000	00	11	1	0	JCC 23	00		READ
018	23	00	01		F5 R2	LTM(AC)	7777	00	11	1	0	JCC 24	00	M to AC	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
019	24	00	01		F1 R1	LMI(R4)	0734	11	11	1	0	JFL 25	11	SAVE D(U)	
														K = TEPDR + 1	
														JFL (EPG17, EPG05)	
020	25	03	01	EPG05	F0 R1	ILR(R8)	0000	00	11	1	0	JCR 00	10	ADDRESS	WRITE
														SIGMA-u-1	
021	25	00	01		F2 R1	SDR(R2)	7777	11	11	1	0	JCC 26	11		
022	26	00	01		F0 R1	ILR(R7)	0000	00	11	1	0	JCC 27	11		
023	27	00	01		F2 R1	SDR(R1)	7777	11	11	1	0	JCC 28	11		
	/* COMPUTE -(U-P)-1 IN R4 AND T(U+1) = T(P) + (U-P) */														
024	28	00	01		F1 R1	LMI(R4)	0000	11	11	1	0	JCC 29	11	ADDRESS OF P	
025	29	00	01		F3 R1	INR(R4)	0000	11	11	1	0	JCC 30	00		READ
026	30	00	01		F5 R2	LTM(AC)	7777	00	11	1	0	JCC 31	00	(P)	READ
027	31	00	01		F1 R1	LMI(R4)	0000	00	11	1	0	JZR 01	11		
028	00	01	01		F7 R3	CMA(AC)	0000	00	11	1	0	JCC 01	00	-(P)-1	READ
029	01	01	01		F0 R2	AMA(AC)	7777	11	11	1	0	JCC 02	00	U-P	READ
030	02	01	01		F1 R1	LMI(R6)	0000	00	11	1	0	JCC 03		ADDRESS OF T(P)	
031	03	01	01		F2 R1	SDR(R4)	7777	00	11	1	0	JCC 04	00	u-p-1	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
032	04	01	01		F5 R2	LTM	7777	00	11	1	0	JCC 05	00		READ
033	05	01	01		F2 R1	SDR	7777	11	11	1	0	JCC 06	11		
034	06	01	01		F0 R1	ILR	0000	00	11	1	0	JCC 07	00		READ
035	07	01	01		F0 R2	AMA(AC)	7777	11	11	1	0	JCC 08	00	T(U + 1) = T(P) + (U-P)	READ
036	07	07	01		F7 R1	CMR(R4)	0000	00	11	1	0	JCC 06	11	(P-U)	
037	06	07	01		F1 R1	LMI	0000	00	11	1	0	JCC 09	11		
038	09	07	01		F7 R3	CMA	0000	00	11	1	0	JCC 10	00		READ
039	10	07	01		F0 R2	AMA	7777	00	11	1	1	JCC 12	00		READ
040	12	07	01		F7 R3	CMA	0000	00	11	1	0	JFL 08	11	JFL(EPG28, EPG27)	
041	08	03	01	EPG27	F0 R1	ILR	0000	00	10	1	0	JCR 07	00		READ
042	08	07	01		F0 R2	AMA	7777	00	11	1	0	JCC 11	00		READ
043	11	07	01		F2 R1	SDR	7777	11	11	1	0	JCR 01	00		READ
044	11	01	01		F5 R2	LTM	7777	11	00	1	0	JCR 08	00		READ
045	08	02	01	EPG28	F1 R1	LMI(R2)	0000	11	11	1	0	JCR 01	11	ADDRESS OF T(U+1)	
046	08	01	01		F3 R1	INR(R1)	0000	11	11	1	0	JCC 12	11	ADDRESS OF SIGMA-u(1)	
047	12	01	01	EPG06	F1 R1	LMI(R1)	0000	11	11	1	0	JCC 13	10	ADDRESS OF SIGMA-u(1)	WRITE

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
048	13	01	01		F3 R1	INR(R4)	0000	11	11	1	0	JCC 14	00		READ
049	14	01	01		F5 R2	LTM(AC)	7777	00	11	1	0	JFL 12	00	JFL (EPG07, EPG08)	READ
050	12	02	01	EPG07	F1 R1	LMI(R2)	0000	11	11	1	0	JCR 01	11	JMP(EPG06)	
	/* COMPARE t(p) - (1-p) to t(u) */														
051	12	03	01	EPG08	F1 R1	DSM(R1)	7777	00	11	1	0	JCR 04	11	NEXT SIGMA (U)	
052	12	04	01		F0 R1	ILR(R6)	0000	00	11	1	0	JCC 15	11	ADDRESS OF T(P)	
053	15	04	01		F2 R1	SDR(R0)	7777	11	11	1	0	JCC 01	11		
054	15	01	01		F3 R1	INR(R0)	0000	11	11	1	0	JCC 16	11	ADDRESS OF SIGMA-P(1)	
055	16	01	01		F1 R1	LMI(R4)	0733	11	11	1	0	JCC 17	11	K = TEPDR ADDRESS OF D(P)	
056	17	01	01		F7 R1	CMR(R9)	0000	00	11	1	0	JCC 18	00	$\overline{T(P)}$ = -T(p)-1	READ
057	18	01	01		F5 R2	LTM(AC)	7777	00	11	1	0	JCC 19	00	D(P)	READ
058	19	01	01		F1 R1	LMI(R4)	0000	00	11	1	0	JCC 20	11		
059	20	01	01		F7 R3	CMA(AC)	0000	00	11	1	0	JCC 21	00	-D(P)-1	READ
060	21	01	01		F0 R2	AMA(AC)	0037	01	11	1	0	JCC 23	00	D(U)-D(P)-1	READ
061	23	01	01		F0 R2	ACM(AC)	0037	00	11	1	0	JCC 24	11	D(U)D(P)-1	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
062	24	01	01		F2 R1	SDR(R4)	7777	11	11	1	0	JCC 25	11		
	/* LOOP TO COMPUTE REMAINING TERMS IN SIGMA-U+1 */														
063	25	01	01	EPG09	F1 R1	LMI(R1)	0000	11	11	1	0	JCC 27	11	ADDRESS OF SIGMA-U(I)	
064	27	01	01		F3 R1	INR(R9)	0000	11	10	1	0	JCC 28	00		READ
065	28	01	01		F7 R3	XNI(AC)	7777	00	11	1	0	JCC 29	00	SIGMA-U D(U)D(P)-1 (SIGMA-P(I)	READ
066	29	01	01		F1 R1	LMI(R2)	0000	11	11	1	0	JCC 30	11	ADDRESS OF SIGMA- (U+1)(I+U-P)	
067	30	01	01		F1 R1	LMI(R0)	0000	11	11	1	0	JCF 31	10	ADDRESS OF SIGMA-P(I) JCF(EPG10, EPG13)	WRITE
068	31	02	01	EPG10	F0 R1	ILR(R4)	0000	00	11	1	0	JCR 01	00	D(U)D(P)-I	READ
069	31	01	01		F5 R2	LTM(T)	7777	00	11	1	0	JCR 04	00	SIGMA-P(I)=0 ?	READ
070	31	04	01		F0 R2	AMA(AC)	7777	00	11	1	0	JFL 26	00	D(U)D(P)-1 SIGMA-P(I) JFL (EPG11, EPG12)	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
071	26	02	01	EPG11	F4 R2	CLA(AC)	0000	00	11	1	0	JCR 01	11	JMP (EPG09)	
072	26	03	01	EPG12	F1 R1	LMI(AC)	0400	00	11	1	0	JCR 04	11	K = TPTCC = 400	
073	26	04	01		F6 R1	NOP(RO)	0000	00	11	1	0	JCC 25	00		READ
074	25	04	01		F5 R2	LTM(AC)	7777	00	11	1	0	JCR 01	00	D(U)D(P)-1	READ
														SIGMA-P(I) - CODE	
														JMP (EPG09)	
075	26	01	01		F1 R1	LMI(R1)	0000	11	11	1	0	JCC 27	11		
	/* DETERMINE WHETHER U SHOULD BECOME P - PREPARE FOR THE CHANGE */														
076	31	03	01	EPG13	F3 R1	INR(R3)	0000	11	11	1	0	JZR 04			
077	00	04	01		F4 R1	CLR(R4)	0000	00	11	1	0	JFL 01	11	JFL (EPG14, EPG91	
078	01	02	01	EPG14	F1 R1	LMI(R4)	0734	11	11	1	0	JCR 04	11	K = TEPDP+1	
														ADDRESS OF D(U)	
079	01	04	01		F6 R1	NOP(R0)	0000	00	11	1	0	JCC 02	00		READ
080	02	04	01		F5 R2	LTM(AC)	7777	00	11	1	0	JCC 03	00	D(U)	READ
081	03	04	01		F1 R1	LMI(R4)	0000	11	11	1	0	JCC 04	11	ADDRESS OF P	
082	04	04	01		F2 R1	SDR(R0)	7777	11	11	1	0	JCC 05	00	D(U) to R(0)	READ
083	05	04	01		F5 R2	LTM(AC)	7777	00	11	1	0	JCC 06	00	P	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
084	06	04	01		F1 R1	LMI(R4)	0000	11	11	1	0	JCC 07	11	ADDRESS OF L(P)	
085	07	04	01		F7 R3	CMA(AC)	0000	00	11	1	0	JCC 08	00	-P-1	READ
086	08	04	01		F0 R2	AMA(AC)	7777	00	11	1	0	JCC 09	00	L(P)-P-1	READ
087	09	04	01		F1 R1	LMI(R4)	0000	11	11	1	0	JCC 10	11	ADDRESS OF U	
088	10	04	01		F2 R1	SDR(R2)	7777	00	11	1	0	JCC 11	00	L(P)-P-2	READ
089	11	04	01		F0 R2	ACM(AC)	0000	11	11	1	0	JCC 13	00	U+1	READ
090	13	04	01		F2 R1	SDR(R1)	7777	00	11	1	0	JCC 14	10	WRITE U+1 SAVE U	WRITE
091	14	04	01		F0 R1	ILR(R2)	0000	00	11	1	0	JCC 16	00	L(P)-P-2	READ
092	16	04	01		F0 R2	AMA(AC)	7777	00	11	1	0	JCC 17	00	L(P)+U-P-1	READ
093	17	04	01		F1 R1	LMI(R4)	0000	00	11	1	0	JCC 18	11	ADDRESS OF L(U)	
094	18	04	01		F7 R3	CMA(AC)	0000	00	11	1	0	JCC 19	00	-(L(P)+U-P)	READ
095	19	04	01		F0 R2	AMA(AC)	7777	00	11	1	1	JCC 20	00		READ
096	20	04	01		F5 R2	LTM(AC)	7777	00	11	1	0	JFL 21	00	L(U)	READ
														JFL (EPG15, EPG16)	
	/* IF L(U) L.T. L(P)+U-P, ROW U BECOMES ROW P; L(U) IS REPLACED W/L(P)+U-P */														
097	21	02	01	EPG15	F2 R1	SDR(R9)	7777	11	11	1	0	JCR 04	11		
098	21	04	01		F0 R1	ILR(R2)	0000	11	11	1	0	JCC 22	11	L(P)+U-P-1 to L(U)	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
099	22	04	01		F3 R3	INA(AC)	0000	11	11	1	0	JCC 23	11		
100	23	04	01		F1 R1	LMI(R4)	7773	00	11	1	0	JCC 24	10	ADDRESS OF D(P)	WRITE
101	24	04	01		F1 R1	LMI(R4)	0000	11	11	1	0	JCC 27	11		
102	27	04	01		F0 R1	ILR(R0)	0000	00	11	1	0	JCC 28	11		
103	28	04	01		F3 R1	INR(R4)	0000	11	11	1	0	JCC 29	10	D(U) REPLACES D(P)	WRITE
104	29	04	01		F1 R1	LMI(R4)	0000	11	11	1	0	JCC 30	11	ADDRESS OF P	
105	30	04	01		F0 R1	ILR(R1)	0000	00	11	1	0	JZR 05	11	U REPLACES P	
106	00	05	01		F1 R1	LMI(R4)	0000	11	11	1	0	JCC 01	10	ADDRESS OF L(P)	WRITE
107	01	05	01		F0 R1	ILR(R9)	0000	00	11	1	0	JCC 02	11		
	/* INTERCHANGE ADDRESS POINTERS OF SIGMA-U AND SIGMA-P THEN SIGMA-U & SIGMA-U+1 */														
108	02	05	01		F0 R1	ILR(R6)	0000	00	11	1	0	JCC 03	10	ADDRESS OF SIGMA-P, L(U) to L(P)	WRITE
109	03	05	01		F2 R1	SDR(R9)	7777	11	11	1	0	JCC 04	11		
110	04	05	01		F0 R1	ILR(R7)	0000	00	11	1	0	JCC 05	11		
111	05	05	01		F2 R1	SDR(R6)	7777	11	11	1	0	JCC 06	11	JMP (EPG16 + 2)	
	/* INTERCHANGE ADDRESS POINTERS OF SIGMA-U AND SIGMA-U+1 */														

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
112	21	03	01	EPG16	F0 R1	ILR(R7)	0000	00	11	1	0	JCR 05	11		
113	21	05	01		F2 R1	SDR(R9)	7777	11	11	1	0	JCC 06	11		
114	06	05	01		F0 R1	ILR(R8)	0000	00	11	1	0	JCC 07	11		
115	07	05	01		F2 R1	SDR(R7)	7777	11	11	1	0	JCC 08	11		
116	08	05	01		F0 R1	ILR(R9)	0000	00	11	1	0	JCC 09	11		
117	09	05	01		F2 R8	SDR(R8)	7777	11	11	1	0	JCC 11	11	JMP (EPG19)	
	/* SHOULD THERE BE ANOTHER PASS? IF SO, INCREMENT U */														
118	25	02	01	EPG17	F1 R1	LMI(AC)	0737	00	11	1	0	JCR 05	10	ADDRESS OF U	WRITE
														K = TEPDR+4	
119	25	05	01		F3 R3	INR(R3)	0000	11	11	1	0	JCC 10	00		READ
120	10	05	01		F0 R2	ACM(AC)	0000	11	11	1	0	JFL 11	00	JFL(EPG18, EPG90)	READ
121	11	02	01	EPG18	F6 R1	NOP(R6)	0000	00	11	1	0	JCR 05	10		WRITE
	/* COMPUTE NEXT D(U) */														
122	11	05	01	EPG19	F0 R1	ILR(R7)	0000	00	11	1	0	JCC 12	11	ADDRESS OF	
														SIGMA-U (I(U))	
123	12	05	01		F2 R1	SDR(R2)	7777	11	11	1	0	JCC 13	11		
124	13	05	01		F0 R1	ILR(R5)	0000	11	11	1	0	JCC 14	11	ADDRESS OF T(U+1)	POWER

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
125	14	05	01		F1 R1	LMI(R2)	0000	11	11	1	0	JCC 15	11		
126	15	05	01		F2 R1	SDR(R4)	7777	11	11	1	0	JCC 16	00		READ
127	16	05	01		F5 R2	LTM(AC)	7777	00	11	1	0	JCC 17	00	LITTLE T(U)	READ
128	17	05	01		F7 R3	CMA(AC)	0000	00	11	1	0	JCC 18	11	-t(U)-1	
129	18	05	01		F1 R1	LMI(R4)	0000	00	11	1	0	JCC 19	11		
130	19	05	01		F2 R1	SDR(R1)	7777	11	11	1	0	JCC 20	00		READ
131	20	05	01		F1 R2	LMM(AC)	0400	00	11	1	0	JCC 22	00	ADDRESS OF T(U+1) (CODE)	READ
132	22	05	01		F6 R1	NOP	0000	00	11	1	0	JCC 23	00		READ
133	23	05	01		F5 R2	LTM(AC)	7777	00	11	1	0	JCC 26	00	T(U+1) (CODE)	READ
134	26	05	01	EPG20	F1 R1	DSM(R4)	7777	00	11	1	0	JCC 24	11		
135	24	05	01		F1 R1	LMI(R2)	0000	11	11	1	0	JCC 31	11	ADDRESS OF SIGMA-U(I)	
136	31	05	01		F2 R1	SDR(R0)	7777	11	11	1	0	JCR 06	00		READ
137	31	06	01		F5 R2	LTM(T)	7777	00	01	1	0	JCC 30	00	SIGMA-U(I) (CODE)	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
138	30	06	01		F1 R2	LMM(AC)	0500	00	11	1	0	JFL 27	00	ADDRESS OF	READ
														SIGMA-U(I) (POWER)	
														JFL(EPG24, EPG21)	
139	27	03	01	EPG21	F3 R1	INR(R1)	0000	11	11	1	0	JCR 06	00		READ
140	27	06	01		F5 R2	LTM(AC)	7777	00	11	1	0	JFL 28	00	SIGMA-U(I)	READ
														(POWER)	
														JFL(EPG22, EPG26)	
141	28	02	01	EPG22	F1 R1	LMI(R4)	0000	00	11	1	0	JCR 05	11	ADDRESS OF	
														T(U+1-I) (POWER)	
142	28	05	01		F6 R1	NOP	0000	00	11	1	0	JCC 27	00		READ
143	27	05	01		F5 R2	LTM(T)	7777	00	11	1	0	JCC 30	00	T(U+1-I) (POWER)	READ
144	30	05	01		F0 R2	AMA(AC)	7777	00	11	1	0	JFL 29	00	SIGMA-U(I)	READ
														T(U+1-I)	
														JFL (EPG25, EPG23)	
145	29	03	01	EPG23	F1 R1	LMI(AC)	0400	00	11	1	0	JCR 06	11		
146	29	06	01		F0 R1	ILR(R0)	1200	00	11	1	0	JCR 05	00	PREVIOUS SUM	READ
147	29	05	01		F7 R3	XNI(AC)	7777	00	11	1	0	JCC 26	00	JMP(EPG20)	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
148	27	02	01	EPG24	F3 R1	INR(R1)	0000	11	11	1	0	JCC 29	11		
149	29	02	01	EPG25	F0 R1	ILR(R0)	0000	00	11	1	0	JFL 24	11	JFL (EPG20, EPG26)	
150	24	02	01		F1 R1	DSM(R4)	7777	00	11	1	0	JCR 05	11		
151	24	03	01	EPG26	F1 R1	LMI(R0)	0500	00	11	1	0	JCC 23	11		
152	28	03	01		F1 R1	LMI(R0)	0500	00	11	1	0	JCC 23	11		
153	23	03	01		F4 R1	CLR(R4)	0000	00	11	1	0	JCR 00	00	JMP(EPG05-2)	READ
	/* CHECK FOR 16-s-2t L.T. 0 */														
154	11	03	01	EPG90	F1 R1	LMI(R7)	0000	00	11	1	0	JCR 06	11	JMP (EPG92)	
155	01	03	01	EPG91	F1 R1	LMI(R8)	0000	00	11	1	0	JCR 06	11		
156	01	06	01		F2 R1	SDR(R7)	7777	11	11	1	0	JCC 13	11		
157	08	06	01		F2 R1	SDR(R7)	7777	11	11	1	0	JCC 11	11		
158	11	06	01	EPG92	F4 R1	CLR(R6)	0000	00	11	1	0	JCC 10	00		READ
159	10	06	01		F5 R2	LTM(AC)	7777	00	10	1	0	JCC 12	00	t	READ
160	12	06	01		F4 R1	CLR(R4)	0000	00	11	1	0	JCC 23	11	JFL (EPG98, EPG93)	
161	23	06	01	EPG93	F1 R1	LMI(R4)	0602	00	11	1	0	JCC 02	11	ADDRESS OF s	
162	02	06	01		F0 R1	ALR(AC)	7777	00	11	1	0	JCC 03	00	2t	READ
163	03	06	01		F0 R2	AMA(AC)	7777	00	11	1	0	JCC 04	00	2t+s	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
164	04	06	01		F1 R3	DCA(AC)	7777	00	11	1	0	JCC 05	11	2t+s L.T.E. 16?	
165	05	06	01		F1 R1	LMI(R6)	0477	00	11	1	0	JCC 06	11	477 = K = SCBIX	
166	06	06	01		F5 R3	TZA(AC)	7760	00	11	1	0	JCC 07	00		READ
167	07	06	01		F5 R2	LTM(AC)	7777	00	11	1	0	JFL 07	00	SCBIX	READ
														JFL (EPG94, EPG97)	
168	07	02	01	EPG94	F4 R1	CLR(R9)		00	11	1	0	JFL 06	11	SET R9 BASED ON	
														SCBIX	
														JFL (EPG96, EPG95)	
169	06	03	01	EPG95	F1 R1	LMI(R9)	0200	00	11	1	0	JCR 02	11		
170	06	02	01	EPG96	F1 R1	LMI(R7)	0000	00	11	1	0	JCF 02	11	ADDRESS OF SIGMA-U	
														JCF (98, 99)	
171	07	03	01	EPG97	F6 R1	NOP	0000	00	11	1	0	JZR 14	11	JUMP TO DECODE	
														FAILURE	
														JMP IOC01	
172	02	02	01	EPG98	F1 R1	LMI(AC)	0602	00	11	1	0	JZR 13	11	JMP RPVAL	
173	02	03	01	EPG99	F0 R1	ILR(R7)	0000	00	10	1	0	JCR 07	00		READ
174	02	07	01		F0 R2	AMA(AC)	7777	00	11	1	0	JCC 01	00		READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
175	01	07	01		F2 R1	SDR(R6)	7777	00	11	1	0	JCC 00	00		READ
176	00	07	01		F0 R2	ACM(AC)	7777	11	11	1	0	JCF 03	00	JMP (EPG89,EPG88)	READ
177	03	02	01	EPG89	F1 R1	LMI(R6)	0000	00	11	1	0	JCR 07	11		
178	03	07	01		F1 R3	DCA(AC)	7777	00	11	1	0	JCC 04	00		READ
179	04	07	01		F5 R2	LTM	7777	00	11	1	1	JCC 05	00		READ
180	05	07	01		F1 R1	DSM(R6)	7777	00	11	1	0	JFL 03	11	JMP (EPG89,EPG88)	
181	03	03	01	EPG88	F1 R1	LMI(R7)	0000	00	11	1	0	JCR 08	11		
182	03	08	01		F0 R1	ILR(R7)	0000	00	11	1	0	JZR 07	10	JMP (EPG89-1)	WRITE

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* SOURCE PROGRAM EPVAL */														
	/* EPGEN LOADS AC WITH ADDRESS OF ERROR POLYNOMIAL TABLE */														
	/* STORE ADDRESS OF ERROR POLYNOMIAL IN SEPEP */														
001	00	07	00	EPVOO	F1 R2	LMM	0644	00	00	1	0	JCC 13	11	644 to MAR, to T	
002	13	07	00		F1 R1	LMI	0000	11	00	1	0	JCC 14	10	AC to MAR, 1+AC to AC	WRITE
003	14	07	00		F2 R1	SDR	7777	11	00	1	0	JCR 14	00	AC to R8	READ
004	14	14	00		F5 R2	LTM	7777	00	00	1	0	JCC 21	00	M to AC, O $\vee$ M to CI	READ
005	21	14	00		F1 R1	DSM	7777	00	00	1	0	JFL 16	11	AC-1 to AC	
	/* READ IEPOO, NUMBER OF ERROR COEFFICIENTS TABLE TEPOL */														
	/* CHECK FOR ZERO ERRORS */														
	/* CHECK FOR ONE ERROR AND TWO ERRORS */														
006	16	11	00	EPV03	F5 R1	TZR	7777	00	00	1	0	JCR 14	11	O $\vee$ AC to CO	
007	16	14	00		F1 R1	DSM	7777	00	00	1	0	JFL 17	11	AC-2 to AC	
008	17	11	00	EPVO5	F5 R1	TZR	7777	00	00	1	0	JCR 14	11	O $\vee$ AC to CO	
009	17	14	00		F1 R1	DSM	7777	00	00	1	0	JFL 18	11	AC-1 to AC	
	/* STORE 1's COMPLEMENT OF (NO.OF ERRORS - 3) IN R3 */														
	/* STORE 1's COMPLEMENT OF NO. OF ERRORS IN R2, R1, R5 */														
010	18	11	00	EPVO7	F1 R3	CIA	0000	00	00	1	0	JCR 14	11	$\overline{AC}$ to AC	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
011	18	14	00		F2 R1	SDR	7777	11	00	1	0	JCC 13	11	AC to R3	
012	13	14	00		F1 R3	CIA	0000	00	00	1	0	JCC 02	11	$\overline{\overline{AC}}$ to AC	
013	02	14	00		F1 R1	LMI	0003	00	00	1	0	JCC 03	11	AC + 3 to AC	
014	03	14	00		F1 R3	CIA	0000	00	00	1	0	JCC 03	11	$\overrightarrow{AC}$ to AC	
015	04	14	00		F2 R1	SDR	7777	11	00	1	0	JCC 11	11	AC to R2	
016	11	14	00		F2 R1	SDR	7777	11	00	1	0	JCC 12	11	AC to R1	
017	12	14	00		F2 R1	SDR	7777	11	00	1	0	JCC 05	11	AC to R5	
	/* CONVERT σ COEFFICIENTS FROM CODE TO POWER */														
	/* SET ADDRESS RO FOR σ POWER - OUTPUT TO TABLE TEVOA */														
	/* READ NEXT σ(CODE) FROM TABLE TEPOL */														
	/* σ CODE = INDEX TO α POWER ADDRESS TABLE TCTPC */														
018	05	14	00		F1 R2	LMM	0660	00	00	1	0	JCC 06	11	660 Ø to AC	
019	06	14	00		F2 R1	SDR	7777	11	00	1	0	JCC 07	11	660 Ø = AC to RO	
020	07	14	00		F1 R1	LMI	0000	11	00	1	0	JCC 08	11	0 $\vee$ R8 to MAR, 1+R8 to R8	
	/* CHECK FOR END OF LOOP I.E. ALL σ s CONVERTED */														
021	08	14	00	EPV08	F3 R1	INR	0000	11	00	1	0	JCC 20	00	1+R2 to R2	READ
022	20	14	00		F1 R2	LMM	0500	00	00	1	0	JFL 19	00	500 $\vee$ M to MAR,	READ
	/* READ σ POWER */														

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* ALL σ'S NOT CONVERTED */														
023	19	10	00	EPV37	F0 R1	ILR	0000	00	00	1	0	JCR 14	00	R2 to AC	READ
024	19	14	00		F5 R2	LTM	7777	00	00	1	0	JCC 09	00	M to AC	READ
	/* WRITE σ POWER IN TABLE TEVOA */														
025	09	14	00		F1 R1	LMI	0000	11	00	1	0	JCC 10	11	$0_{\vee}$ R0 to MAR, 1+RO to R0	
026	10	14	00		F1 R1	LMI	0000	11	00	1	0	JCC 08	10	$0_{\vee}$ R8 to MAR, 1+R8 to R8	WRITE
	/* SOLUTION FOR MORE THAN TWO ROOTS */														
	/* INITIALIZE ERASURE POSITION CHECK */														
	/* READ NUMBER OF ERASURES FROM CURRENT DATA BLOCK */														
027	19	11	00	EPV09	F0 R1	ILR	0000	00	00	1	0	JCR 13	11	R9 to AC	
028	19	13	00		F1 R1	LMI	0070	11	00	1	0	JCC 18	11	70Ø to MAR if R9=0 270Ø to MAR if R9=200Ø 71Ø to AC if R9=0 271Ø to AC if R9=200Ø	
	/* STORE NEXT ERASURE ADDRESS */														
029	18	13	00		F2 R1	SDR	7777	11	00	1	0	JCC 17	00	AC to R4	READ
030	17	13	00		F0 R2	AMA	7777	00	00	1	0	JCC 16	00	M + AC to T	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* WRITE 37Ø in ADDRESS FOLLOWING LAST ERASURE IN TADIØ/TBDIØ */														
031	16	13	00		F1 R2	LMM	0037	00	00	1	0	JCC 20	11	K to AC, K to MAR	
032	20	13	00		F1 R1	LMI	0000	00	00	1	0	JCC 21	11	T to MAR	
033	21	13	00		F1 R1	LMI	0000	11	00	1	0	JCC 22	10	R4 to MAR, 1+R4 to R4	WRITE
	/* READ FIRST ERASURE, STORE IN R8 */														
034	22	13	00		F4 R1	CLR	0000	00	00	1	0	JCC 23	00	0 to R6	READ
035	23	13	00		F0 R2	ACM	0000	00	00	1	0	JCC 24	00	M to AC	READ
036	24	13	00		F2 R1	SDR	7777	11	00	1	0	JCC 25	11	AC to R8	
	/* SET ROW INDEX TO 111 111 100 000 */														
037	25	13	00		F1 R1	LMI	7740	00	00	1	0	JCC 26	11	7740Ø to R6	
	/* SET INDEX FOR ROOT OUTPUT TEVAL */														
038	26	13	00		F1 R2	LMM	0647	00	00	1	0	JCC 27	11	647 Ø to AC	
039	27	13	00		F2 R1	SDR	7777	11	00	1	0	JCC 28	11	647 Ø to R9	
	/* BEGIN NEW ROW */														
	/* UPDATE ROW INDEX. CHECK FOR LAST ROW */														
	/* STORE 5 LSD OF ROW INDEX IN COLUMN INDEX R7 */														
040	28	13	00	EPV10	F0 R1	ILR	0000	11	00	1	0	JCC 29	11	1 + R6 to R6, AC	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
041	29	13	00		F2 R1	CSR	0037	11	00	1	0	JFL 21	11	AC (5LS D) to R7	
	/* ROWS REMAIN FOR PROCESSING */														
	/* CHECK IF ROW INDEX (5LSD) = COMPLEMENT OF CURRENT ERASURE */														
	/* SET ROW CUMMULATIVE SUM TO 0001 */														
042	21	10	00	EPV11	F2 R1	CSR	0000	11	00	1	0	JCR 12	11	AC (5LS D) to T	
043	21	12	00		F4 R1	CLR	0000	00	00	1	0	JCC 22	11	0 to R2	
044	22	12	00		F0 R1	ILR	0000	00	00	1	0	JCC 20	11	R 8 to AC	
045	20	12	00		F7 R1	XNR	7777	00	00	1	0	JCC 19	11	R7 $\overline{\oplus}$ R8 = T $\overline{\oplus}$ AC to T	
046	19	12	00		F5 R3	TZA	0037	00	00	1	0	JCC 18	11	0_V T to CI	
047	18	12	00		F3 R1	INR	0000	11	00	1	0	JFL 24	11	1 + R2 to R2	
	/* ROW INDEX = ERASURE COMPLEMENT */														
	/* READ NEXT ERASURE */														
048	24	10	00	EPV12	F1 R1	LMI	0000	11	00	1	0	JCR 12	11	R4 to MAR, 1 + R4 to R4	
049	24	12	00		F6 R1	NOP	0000	00	00	1	0	JCC 25	00	R0 to R0	READ
050	25	12	00		F5 R2	LTM	7777	00	00	1	0	JCC 28	00	M to AC	READ
051	28	12	00		F2 R1	SDR	7777	11	00	1	0	JCR 13	11	AC to R8	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/WRITE	MICROFUNCTION	
	/* INITIALIZE ADDRESS OF σ COEFFICIENT IN TEVOA */														
052	24	11	00	EPV13	F5 R1	CLR	0660	00	00	1	0	JCR 09	11	$K_\wedge R0 = 660\emptyset$ to Ro	
	/* INITIALIZE INDEX R1. R5 to R1 */														
053	24	09	00		F0 R1	ILR	0000	00	00	1	0	JCC 23	11	R5 to AC	
054	23	09	00		F2 R1	SDR	7777	11	00	1	0	JCC 22	11	AC = R5 to R1	
	/* INCREMENT NO. OF σ TERMS - CHECK ALL σ TERMS USED */														
	/* READ NEXT σ TERM - CHECK σ = ZERO */														
	/* COMPUTE $\sigma . (\alpha^x)$ */														
055	22	09	00	EPV14	F0 R1	ILR	0000	11	00	1	0	JCC 21	11	1 + R1 to R1, AC	
056	21	09	00		F1 R1	LMI	0000	11	00	1	0	JFL 25	11	RO to MAR, 1+RO to RO	
	/* σ TERMS REMAIN */														
057	25	10	00	EPV15	F0 R1	ILR	0000	00	00	1	0	JCR 09	00	R7 to AC	READ
058	25	09	00		F5 R2	LTM	7777	00	00	1	0	JCC 20	00	$0 \vee$ M to CO, M to T	READ
059	20	09	00		F0 R1	ALR	7777	00	00	1	0	JFL 26	11	T + AC to T, AC	
	/* $\sigma \neq$ ZERO */														
	/* CONVERT $\sigma . (\alpha^x)$ TO CODE, READ FROM TPTCC */														
060	26	11	00	EPV16	F1 R1	LMI	0400	00	00	1	0	JCR 12	11	$400 \vee$ AC to MAR to AC	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
061	26	12	00		F0 R1	ILR	0000	00	00	1	0	JCC 23	00	R2 to AC	READ
062	23	12	00		F7 R3	XNI	7777	00	00	1	0	JCR 10	00	AC $\oplus$ I($\overline{M}$) to AC	READ
	/* CUMMULATIVE ADD (XOR) SUM TO ROW SUM R2 */														
063	23	10	00		F2 R1	SDR	7777	11	00	1	0	JCC 26	11	AC to R2	
	/* GALOIS MULTIPLY (ADD) ROW α TERM BY COLUMN α TERM */														
	/* ADJUST POWER AND STORE IN COLUMN α TERM */														
064	26	10	00	EPV17	F0 R1	ILR	0000	00	00	1	0	JCR 09	11	R6 to AC	
065	26	09	00		F0 R1	ALR	0037	00	00	1	0	JCC 19	11	R7 +$\overset{K}{A}$ to R7, AC	
066	19	09	00		F3 R3	AIA	0037	00	00	1	0	JCC 18	11	AC + I (D) to AC	
067	18	09	00		F2 R1	SDR	0037	11	00	1	0	JCC 22	11	AC to R7	
	/* CHECK IF ROOT FOUND, SUM = 0 */														
068	25	11	00	EPV18	F5 R1	TZR	7777	00	00	1	0	JCR 14	11	$O_V R_2$ to CO	
069	25	14	00		F0 R1	ILR	0000	00	00	1	0	JFL 29	11	R6 to AC	
	/* ROOT FOUND - WRITE INVERSE ROOT POWER IN TEVAL */														
070	29	10	00	EPV19	F1 R1	LMI	0000	11	00	1	0	JCR 12	11	R9 to MAR, 1+R9 to R9	
071	29	12	00		F1 R3	CIA	0000	00	00	1	0	JCC 17	11	$\overline{AC}$ to AC 5LSD	
	/* STORE INVERSE ROOT IN REGISTER T */														
	/* CONVERT INVERSE ROOT TO CODE */														

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
072	17	12	00		F1 R1	LMI	0400	00	00	1	0	JCC 16	10	$400 \vee$ AC to MAR to AC	WRITE
073	16	12	00		F2 R1	CSR	0077	11	00	1	0	JCR 09	00	$\overset{K}{AC}$ to R1	READ
074	16	09	00		F5 R2	LTM	7777	00	00	1	0	JCC 17	00	M to AC	READ
	/* READ CUMMULATIVE INVERSE ROOT SUM CODE */														
	/* XOR WITH NEW ROOT. STORE IN SIRCS */														
075	17	09	00		F1 R2	LMM	0670	11	00	1	0	JCR 08	11	670Ø to MAR, 671Ø to T	
076	17	08	00		F6 R0	NOP	0000	00	00	1	0	JCC 16	00	R0 to R0	READ
077	16	08	00		F7 R3	XNI	7777	00	00	1	0	JCC 18	00	AC $\overline{\oplus}$ I $(\overline{M}) \wedge$ K to AC	READ
	/* READ CUMMULATIVE INVERSE ROOT PRODUCT POWER */														
	/* ADD TO NEW ROOT */														
078	18	08	00		F1 R1	LMI	0000	00	00	1	0	JCC 19	10	$0 \vee T =$ 671Ø to MAR	WRITE
079	19	08	00		F0 R1	ILR	0000	00	00	1	0	JCC 20	00	R1 to AC	READ
080	20	08	00		F0 R2	AMA	7777	00	00	1	0	JCC 21	00	M + AC to AC	READ
	/* ADJUST FOR POWER >31 STORE RESULT IN SIRPS */														
081	21	08	00		F3 R3	INA	0037	00	00	1	0	JCC 22	11	AC + I (D) to AC	
082	22	08	00		F2 R2	CSA	0037	11	00	1	0	JCC 23	11	$\overset{K}{AC}$ to AC	
083	23	08	00		F1 R1	LMI	0000	00	00	1	0	JCC 28	11	$0 \vee T =$ 671Ø to MAR	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* INCREMENT ROOT LIMIT R3 */														
	/* CHECK FOR ALL ROOTS MINUS 2 FOUND (LIMIT = -3) */														
084	24	08	00		F0 R1	ILR	0000	11	00	1	0	JCC 25	10	R3 + 1 to R3, AC	WRITE
085	25	08	00		F0 R1	ILR	0000	11	00	1	0	JFL 27	11	1+R6 to R6, AC	
	/* ROOTS - 2 NOT ALL FOUND */														
086	27	10	00	EPV50	F2 R1	CSR	0037	11	00	1	0	JFL 21	11	AC (5LSD) to R7	
	/* NO ROOT - ON THIS ITERATION */														
087	29	11	00	EPV38	F0 R1	ILR	0000	11	00	1	0	JCR 13	11	1+R6 to R6, AC	
088	21	11	00	EPV39	F6 R1	NOP	0000	11	00	1	0	JZR 14 P 01	11	R0 to R0 JMP IOC01	
	/* ZERO ROOTS */														
	/* WRITE 0 FOR NUMBER OF ROOTS IN TEVAL */														
089	16	10	00	EPV02	F4 R1	CLR	0000	00	00	1	0	JCR 07	11	0 to AC	
090	16	07	00		F1 R2	LMM	0646	00	00	1	0	JCC 15	11	646Ø to MAR, to T	
091	15	07	00		F6 R1	NOP	0000	00	00	1	0	JZR 13 P 01	10		WRITE
	/* ONE ROOT */														
	/* WRITE 1 FOR NUMBER OF ROOTS IN TEVAL (646) */														
092	17	10	00	EPV04	F0 R2	ACM	0000	11	00	1	0	JCR 07	11	1 to AC	
093	17	07	00		F1 R2	LMM	0646	00	00	1	0	JCC 18	11	646 to MAR, to T	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
094	18	07	00		F1 R1	LMI	0000	11	00	1	0	JCC 19	10	R8 to MAR, 1+R8 to R8	WRITE
	/* CONVERT σ COEFFICIENT IN CODE FORM TO POWER FORM */														
	/* CHECK IF POWER = 31D = 37 Ø */														
095	19	07	00		F6 R1	NOP	0000	00	00	1	0	JCC 20	00		READ
096	20	07	00		F1 R2	LMM	0500	00	00	1	0	JCC 21	00	500_v M to MAR, 500+M to AC	READ
097	21	07	00		F6 R1	NOP	0000	00	00	1	0	JCC 22	00		READ
098	22	07	00		F6 R3	ORI	0037	00	00	1	0	JCC 01	00	0_v I to CO, to T	READ
099	01	07	00		F5 R2	LTM	7777	00	00	1	0	JFL 02	00	M to AC	READ
	/* σ POWER = 0 */														
100	02	02	00	EPV39	F4 R1	CLR	0000	00	00	1	0	JCR 03	11	0 to AC	
	/* σ POWER ≠ 0 */														
101	02	03	00	EPV38	F1 R2	LMM	0647	00	00	1	0	JCR 07	11	647 to MAR to T	
102	02	07	00		F6 R1	NOP	0000	00	00	1	0	JZR 06 P	10 01	JMP RPG00	WRITE
	/* TWO ROOTS */														
	/* READ σ COEFFICIENT 1ST DEGREE TERM - CODE FORM */														
103	18	10	00	EPV01	F1 R1	LMI	0000	11	00	1	0	JCR 06	11	R8 to MAR, 1+R8 to R8	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* STORE ZERO IN INVERSE ROOTS SUM CODE R2 */														
104	18	06	00		F4 R1	CLR	0000	00	00	1	0	JCC 26	00	0 to R2	READ
105	26	06	00		F5 R2	LTM	7777	00	00	1	0	JCR 08	00	0_v M to C0, M to AC	READ
	/* CHECK σ COEFFICIENT FOR ZERO VALUE */														
	/* STORE ZERO IN INVERSE ROOTS PRODUCT POWER R3 */														
106	26	08	00		F4 R1	CLR	0000	00	00	1	0	JFL 28	11	0 to R3	
	/* σ COEFFICIENT NOT ZERO */														
	/* STORE σ IN R6. STORE ONES COMPLEMENT OF NO. OF ROOTS, IE. 2 IN R5*/														
107	28	11	00		F2 R1	SDR	7777	11	00	1	0	JCR 09	11	AC to R6	
108	28	09	00		F2 R1	CSR	0000	00	00	1	0	JCC 29	11	- 1 = 1—1 to AC	
	/* READ σ COEFFICIENT OF 2ND DEGREE TERM */														
	/* CONVERT CODE TO POWER. STORE IN R7 */														
	/* SET OUTPUT ROOT INDEX FOR TEVAL IN R9 */														
109	29	09	00		F1 R1	LMI	0000	00	00	1	0	JCC 30	11	R8 to MAR	
110	30	09	00		F2 R1	SDR	7775	11	00	1	0	JCC 31	00	7775 to R5	READ
111	31	09	00		F1 R2	LMI	0500	00	00	1	0	JCC 27	00	500_v M to MAR; 500+M to T	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
112	27	09	00		F2 R1	CSR	0647	11	00	1	0	JCR 08	00	647Ø to R9	READ
113	27	08	00		F5 R2	LTM	7777	00	00	1	0	JCC 28	00	M to AC	READ
114	28	08	00		F2 R1	SDR	7777	11	00	1	0	JCR 07	11	AC to R7	
/* DECODE FAILURE */															
115	28	10	00	EPV40	F6 R1	NOP	0000	00	00	1	0	JZR 14 P 01	11	RO to RO JMP IOC01	
/* READ CUMMULATIVE INVERSE ROOTS CODE SUM SIRCS, STORE IN R_2 */															
116	27	11	00	EPV20	F1 R2	LMM	0670	00	00	1	0	JCR 07	11	0_v670Ø to MAR, 670Ø to T	
117	27	07	00		F6 R1	NOP	0000	00	00	1	0	JCC 26	00	0 to R_0	READ
118	26	07	00		F5 R2	LTM	7777	00	00	1	0	JCC 12	00	M to AC	READ
119	12	07	00		F1 R2	LMM	0671	00	00	1	0	JCC 11	11	671Ø to MAR, 671Ø to T	
/* READ CUMMULATIVE INVERSE ROOTS POWER PRODUCT SIRPS to R_3 */															
120	11	07	00		F2 R1	SDR	7777	11	00	1	0	JCC 10	00	AC to R_2	READ
121	10	07	00		F5 R2	LTM	7777	00	00	1	0	JCC 09	00	M to AC	READ
/* READ ERROR POLYNOMIAL ADDRESS SEPEP = 644 */															
/* READ 1st σ COEFFICIENT CODE FORM */															
/* STORE IN R_6 */															
122	09	07	00		F1 R2	LMM	0644	00	00	1	0	JCC 08	11	644Ø to MAR to T	
123	08	07	00		F2 R1	CSR	7777	11	00	1	0	JCC 07	00	AC to R_3	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
124	07	07	00		F5 R2	LTM	0037	00	00	1	0	JCC 25	00	K M to AC	READ
125	25	07	00		F0 R1	ILR	0000	11	00	1	0	JCC 06	11	1 + AC to AC	
126	06	07	00		F1 R1	LMI	7000	00	00	1	0	JCC 05	11	700_{V}AC to MAR,	
														700 + AC to AC	
127	05	07	00		F0 R1	ILR	0000	00	00	1	0	JCC 04	00	R_5 to AC	READ
128	04	07	00		F5 R2	LTM	7777	00	00	1	0	JCC 03	00	M to T	READ
	/* READ σ MAX. COEFFICIENT (USE R_5) POWER FORM */														
	/* STORE IN R_7 */														
129	03	07	00		F1 R3	CIA	0000	00	00	1	0	JCC 23	11	$\overline{AC}$ to AC (R_5)	
	/* ADD INDEX TO TEVOA ADDRESS 657Ø */														
130	23	07	00		F1 R1	LMI	0657	00	00	1	0	JCC 24	11	AC+657Ø to AC	
131	24	07	00		F1 R1	LMI	0000	00	00	1	0	JCC 30	11	O_{V}AC to MAR	
132	30	07	00		F0 R1	ILR	0000	00	00	1	0	JCR 06	00	T to AC (σ CODE)	READ
133	30	06	00		F2 R1	SDR	7777	11	00	1	0	JCC 17	00	AC to R_6	READ
134	17	06	00		F5 R2	LTM	7777	00	00	1	0	JCC 28	00	M to AC σ POWER MAX	READ
135	28	06	00		F2 R1	SDR	7777	11	00	1	0	JCR 07	11	AC to R_7	
	/* ADD ONE'S COMPLEMENT OF σ MAX AND */														
	/* INVERSE ROOTS POWER PRODUCT (R_3) */														

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION
	/* CHECK IF σMAX < INVERSE ROOTS POWER SUM */													
136	28	07	00	EPV26	F1 R3	CIA	0000	00	00	1	0	JCC 28	11	$\overline{AC}$ to AC = $\overline{\sigma}$MAX
137	29	07	00		F2 R1	SDR	7777	11	00	1	0	JCR 08	11	AC to R_4
138	29	08	00		F0 R1	ILR	0000	00	00	1	0	JCR 06	11	R_3 to AC
139	29	06	00		F0 R1	ALR	7777	00	00	1	0	JCC 27	11	R_4 + AC to R_4, AC
140	27	06	00		F1 R3	CIA	0000	00	00	1	0	JFL 24	11	$\overline{AC}$ to AC
	/* σ MAX $\geqslant$ INVERSE ROOTS POWER PRODUCT */													
141	24	02	00	EPV21	F2 R1	SDR	7777	11	00	1	0	JCR 06	11	AC to R_4 = P
	/* σ MAX < INVERSE ROOTS POWER PRODUCT ($\neq$ 0) */													
142	24	03	00	EPV22	F1 R1	LMI	0037	00	00	1	0	JCR 05	11	R_7 + 37Ø to R_7
143	24	05	00		F0 R1	ILR	0000	00	00	1	0	JCC 20	11	R_3 to AC
144	20	05	00		F1 R3	CIA	0000	11	00	1	0	JCR 06	11	1 + $\overline{AC}$ to AC
145	20	06	00		F0 R1	ALR	7777	00	00	1	0	JCC 09	11	R_7 + AC to AC = P
146	09	06	00		F2 R1	SDR	7777	11	00	1	0	JCC 24	11	AC to R_4 = P
	/* READ CURRENT BLOCK INDEX INDICATOR SCBIX */													
	/* ESTABLISH CURRENT BLOCK ADDRESS IN R_8 */													
147	24	06	00		F1 R2	LMM	0477	00	00	1	0	JCC 15	11	477Ø to MAR to AC
148	15	06	00		F4 R1	CLR	0000	00	00	1	0	JCC 14	00	0 to R_8

READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
149	14	06	00		F5 R2	LTM	7777	00	00	1	0	JCC 16	00	O_VM to CO, M to AC	READ
150	16	06	00		F1 R1	LMI	0004	00	00	1	0	JFL 28	11	4 to MAR, 4 to R_8	
	/* CURRENT BLOCK IS A, 0 to R_8 */														
	/* CURRENT BLOCK IS B, 204∅ to R_8 */														
151	28	03	00	EPV41	F1 R1	LMI	0200	00	00	1	0	JCR 02	11	204∅ to MAR, 204∅ to R_8	
	/* READ CODE SELECT FOR CURRENT BLOCK */														
	/* CHECK FOR LONG OR SHORT CODE */														
152	28	02	00	EPV42	F0 R1	ILR	0000	00	00	1	0	JCR 05	00	R_4 = P to AC	READ
153	28	05	00		F5 R2	TZM	0001	00	00	1	0	JCC 29	00	$O_V\overset{K}{M}$ to CO, T	READ
154	29	05	00		F2 R1	SDR	7777	00	00	1	0	JFL 25	11	P-1 to R_0	
	/* LONG CODE */														
155	25	02	00	EPV23	F4 R1	CLR	0000	00	00	1	0	JCC 04	11	0 to R_1	
156	04	02	00		F1 R1	LMI	0000	11	00	1	0	JZR 06	11	1 to R_1, 0 to MAR	
	/* SHORT CODE, CHECK IF $P < 9$ */														
157	25	03	00	EPV24	F1 R3	CIA	0000	00	00	1	0	JCR 05	11	$\overline{AC} = \overline{P}$ to AC	
158	25	05	00		F1 R1	L MI	0011	00	00	1	0	JCC 26	11	$\overline{P}$ + 11∅ to AC	
159	26	05	00		F4 R1	CLR	0000	00	00	1	0	JFL 23	11	0 to R_0	
	/* STORE UPPER AND LOWER LIMITS FOR $P < 9$ */														

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION
160	23	03	00		$\overline{F1}$ R1	LMI	0023	00	00	1	0	JCR 05	11	0 + 23Ø to R_0, 23Ø to MAR
161	23	05	00		F0 R1	ILR	0000	00	00	1	0	JCC 10	11	R_4 to AC = P
162	10	05	00		F1 R1	LMI	0014	00	00	1	0	JCC 11	11	AC + 14Ø to AC
163	11	05	00		F2 R1	SDR	7777	11	00	1	0	JZR 06	11	AC to R_1
	/* FOR P $\geqslant$9 CHECK IF p - 4 < 20 */													
164	23	02	00	EPV25	F0 R1	ILR	0000	00	00	1	0	JCR 06	11	R_4 to AC = P
165	23	06	00		F1 R1	LMI	7774	00	00	1	0	JCC 13	11	AC + 7774 to AC
166	13	06	00		F1 R3	CIA	0000	00	00	1	0	JCC 11	11	$\overline{AC}$ to AC
167	11	06	00		F1 R1	LMI	0024	00	00	1	0	JCC 25	11	$\overline{(P-4)}$ AC + 24Ø to AC
168	25	06	00		F0 R1	ILR	0000	00	00	1	0	JFL 22	11	R_4 to AC = P
	/* STORE UPPER AND LOWER LIMITS FOR P - 4 < 20 P $\geqslant$ 9 */													
169	22	03	00	EPV26	F1 R1	LMI	7774	00	00	1	0	JCR 05	11	P - 4 to AC
170	22	05	00		F2 R1	SDR	7777	11	00	1	0	JCC 27	11	AC to R_0
171	27	05	00		F4 R1	CLR	0000	00	00	1	0	JCC 09	11	0 to R_1
172	09	05	00		F1 R1	LMI	0004	00	00	1	0	JZR 06	11	4 to R_1
	/* STORE UPPER AND LOWER LIMITS FOR P $\geqslant$9 AND P - 4 $\geqslant$20 */													
173	22	02	00	EPV27	F1 R1	LMI	7755	00	00	1	0	JCR 06	11	P - 19 to AC

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION
174	22	06	00		F2 R1	SDR	7777	11	00	1	0	JCC 10	11	AC to R_1
175	10	06	00		$\overline{F1}$ R1	LMI	0023	00	00	1	0	JZR 06	11	23Ø to R_0

/* FIND A PAIR OF VALUES WITH SUM = P */

/* AND WITH XOR TO σ_1 CODE */

/* STORE LONG/SHORT CODE INDICATOR IN R_3 */

/* CHECK IF UPPER LIMIT OF P >LOWER LIMIT OF P */

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION
174	00	06	00	EPV28	F0 R1	ILR	0000	00	00	1	0	JCC 02	11	T to AC
175	02	06	00		F2 R1	SDR	7777	11	00	1	0	JCC 01	11	AC to R_3
176	01	06	00		F4 R1	CLR	0000	00	00	1	0	JCR 03	11	0 to R_7
177	01	03	00		F0 R1	ILR	0000	00	00	1	0	JCR 02	11	R_1 to AC
178	01	02	00		F1 R3	CIA	0000	00	00	1	0	JCC 06	11	$\overline{AC}$ to AC
179	06	02	00		F2 R2	SDA	7777	11	00	1	0	JCR 06	11	AC to T
180	06	06	00		F0 R1	ILR	0000	00	00	1	0	JCC 07	11	R_0 to AC
181	07	06	00		F0 R1	ALR	7777	00	00	1	0	JCC 05	11	T + AC to AC
182	05	06	00		F1 R1	LMI	0400	00	00	1	0	JFL 12	11	0_v R_o to MAR

/* UPPER LIMIT > LOWER LIMIT */

/* CONVERT UPPER AND LOWER LIMITS FROM POWER TO CODE */

/* XOR THE RESULTS WITH THE CUMMULATIVE CODE SUM IN R_2 */

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
183	12	03	00	EPV29	F0 R1	ILR	0000	00	00	1	0	JCR 06	00	R_2 to AC	READ
184	12	06	00		F5 R2	LTM	7777	00	00	1	0	JCC 19	00	M to T	READ
185	19	06	00		F1 R1	LMI	0400	00	00	1	0	JCC 04	11	0_V R_1 to MAR	
186	04	06	00		F1 R3	CIA	0000	00	00	1	0	JCC 31	00	$\overline{AC}$ $(\overline{R_2})$ to AC	READ
187	31	06	00		F7 R3	XNI	7777	00	00	1	0	JCR 10	00	T $\overline{\oplus}$ $(I_\wedge K)$ to T	READ
188	31	10	00		F7 R1	XNR	7777	00	00	1	0	JCR 14	11	T $\overline{\oplus}$ AC to T	
189	31	14	00		F1 R3	CIA	0000	00	00	1	0	JCR 11	11	$\overline{T}$ to T	
	/* DOES RESULT EQUAL CODE OF σ_1 IN R_6 */														
190	31	11	00		F0 R1	ILR	0000	00	00	1	0	JCR 12	11	R_6 to AC	
191	31	12	00		F7 R1	XNR	7777	00	00	1	0	JCR 13	11	T $\overline{\oplus}$ AC to T	
192	31	13	00		F5 R1	TZR	7777	00	00	1	0	JCR 07	11	0_V T to CO	
193	31	07	00		F5 R1	LDR	0037	00	00	1	0	JFL 21	11	R_0 to R_0	
	/* XOR RESULT $\neq$ σ_1 CODE VALUE IN R_6 */														
	/* SUBTRACT 1 FROM UPPER LIMIT. ADD 1 TO LOWER LIMIT */														
194	21	03	00	EPV30	F1 R1	DSM	7777	00	00	1	0	JCC 05	11	$R_0 - 1$ TO $R_0 \vdash\!\dashv$ to MAR	
195	05	03	00		F5 R1	LDR	0037	00	00	1	0	JCR 02	11	K R_1 to R_1	
196	05	02	00		F0 R1	ILR	0000	11	00	1	0	JCC 01	11	$R_1 + 1$ to R_1 to AC	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION
	/* UPPER LIMIT ⩽ LOWER LIMIT */													
	/* CHECK FOR LONG OR SHORT CODE */													
197	12	02	00	EPV31	F5 R1	TZR	0003	00	00	1	0	JCR 05	11	$0 \vee^{K} R_3$ to CO
198	12	05	00		F5 R1	TZR	7777	00	00	1	0	JFL 08	11	$0 \vee R_7$ to CO
	/* LONG CODE */													
	/* CHECK IF 2ND SET OF LIMITS HAVE BEEN TRIED */													
199	08	02	00	EPV32	F4 R1	CLR	0000	00	00	1	0	JFL 07	11	0 to R_0
	/* TRY SECOND SET OF LIMITS */													
	/* SET 2ND SET OF LIMITS INDICATOR */													
200	07	02	00	EPV33	F1 R1	LMI	0036	00	00	1	0	JCC 15	11	R_0+36Ø to R_0, 36Ø to MAR
201	15	02	00		F0 R1	ILR	0000	11	00	1	0	JCR 03	11	$R_4 + 1$ to R_4, AC
202	15	03	00		F2 R1	SDR	7777	11	00	1	0	JCC 06	11	AC to R_1
203	06	03	00		F0 R1	ILR	0000	11	00	1	0	JCC 01	11	$R_7 + 1$ to R_7, AC
204	08	03	00		F6 R1	NOP	0000	00	00	1	0	JZR 14	11	R_0 to R_0 JMP IOC01
205	07	03	00		F6 R1	NOP	0000	00	00	1	0	JZR 14	11	R_0 to R_0 JMP IOC01
	/* ROOTS FOUND */													
	/* CHECK IF UPPER LIMIT = 37Ø */													

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
206	21	02	00	EPV34	F1 R2	LMM	7740	00	00	1	0	JCR 06	11	7740Ø to T	
207	21	06	00		F0 R1	ILR	0000	00	00	1	0	JCC 08	11	R_0 to AC	
208	08	06	00		F7 R1	XNR	7777	00	00	1	0	JCR 05	11	$T \overline{\oplus} AC$ to T	
209	08	05	00		F5 R1	TZR	7777	00	00	1	0	JCC 02	11	0_V T to CI	
210	02	05	00		F0 R1	ILR	0000	00	00	1	0	JFL 03	11	R_0 to AC	
	/* UPPER LIMIT = 37Ø */														
211	03	02	00	EPV35	F0 R1	ILR	0000	00	00	1	0	JCR 06	11	R_1 to AC	
212	03	06	00		F5 R1	CLR	0000	00	00	1	0	JCR 03	11	0 to R_1	
	/* ROOTS FOUND. WRITE LOWER LIMIT IN NEXT TEVAL ADDRESS */														
213	03	03	00	EPV36	F1 R1	LMI	0000	11	00	1	0	JCR 05	11	R_9 to MAR, R_9+1 to R_9	
214	03	05	00		F0 R1	ILR	0000	00	00	1	0	JCC 21	10	R_1 to AC	WRITE
215	21	05	00		F1 R1	LMI	0000	11	00	1	0	JCC 07	11	R_9 to MAR, R_9+1 to R_9	
	/* WRITE NUMBER OF ROOTS IN TEVAL (646Ø RAM) */														
216	07	05	00		F0 R1	ILR	0000	00	00	1	0	JCC 06	10	R_5 to AC	WRITE
217	06	05	00		F1 R3	CIA	0000	00	00	1	0	JCC 05	11	$\overline{AC}$ to AC	
218	05	05	00		F1 R2	LMM	0646	00	00	1	0	JCC 04	11	646Ø to MAR, to T	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
219	04	05	00		F0 R1	ILR	0000	00	00	1	0	JCC 01	10	R_8 to AC	WRITE
	/* RESTORE CURRENT BLOCK INDICATOR TO R_9 */														
220	01	05	00		F2 R1	CSR	0770	11	00	1	0	JZR 06 P 01	11	K AC to R_9	
	/* EXIT TO RPGEN */														

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* SOURCE PROGRAM RPGEN */														
001	00	06	01	RPG00	F4 R2	CIA(AC)	0000	00	00	1	0	JCC 09	11		
002	09	06	01		F1 R1	LMI(AC)	0644	00	11	1	0	JCC 13	11	1st SIGMA-e	
														644 to MAR 645 to AC	
003	13	06	01		F6 R1	NOP(RO)	0000	00	11	1	0	JCC 14	00		READ
004	14	06	01		F1 R2	LMM(AC)	0700	11	11	1	0	JCC 15	00	M + 700 to MAR	READ
														M + 700 + 1 to AC	
005	15	06	01		F2 R1	SDR(RO)	7777	11	11	1	0	JCC 16	11	AC to R0	
006	16	06	01		F4 R1	CLR(R2)	0000	00	11	1	0	JCC 17	00	0 to R2	READ
007	17	06	01		F5 R2	LTM(AC)	7777	00	11	1	0	JCC 18	00	t INTO AC	READ
008	18	06	01		F1 R1	LMI(R2)	0561	00	11	1	0	JCC 19	11	POINT TO O/P AREA	
														561 to MAR	
009	19	06	01		F2 R1	SDR(R8)	7777	11	11	1	0	JCC 20	00	t INTO R8	READ
														AC = t to R8	
010	20	06	01		F0 R2	AMA(AC)	7777	00	11	1	0	JCC 21	00	s+t INTO AC	READ
011	21	06	01		F2 R1	SDR(R6)	7777	11	11	1	0	JCC 22	11	s + t INTO R6	
012	22	06	01		F7 R1	CMR(R6)	0000	00	11	1	0	JCR 07	11	$\overline{s+t}$ INTO R6	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
013	22	07	01		F7 R1	CMR(R8)	0000	00	11	1	0	JFL 16	10	s+t INTO O/P AREA	WRITE
														$\overline{t}$ to R8	
	/* PREPARE FOR A PASS THROUGHOUT THE LOOP WITH A NEW (S-SIGMA(E) (T) COEFFICIENT */														
014	16	02	01	RPG01	F3 R1	INR(R8)	0000	11	11	1	0	JCR 07	11	INC T COUNTER 1 + R8 to R8	
015	16	07	01		F0 R1	ILR(R6)	0000	11	11	1	0	JFL 17	11	INC $\overline{s+t}$ COUNTER AND	
														STORE IN R7	
016	17	03	01	RPG02	F4 R1	CLR(R2)	0000	00	11	1	0	JZR 13	11	0 to R2	
017	17	02	01	RPG03	F2 R1	SDR(R7)	7777	11	11	1	0	JCR 07	11	$\overline{s+t}$ + 1 to R7	
018	17	07	01		F0 R1	ILR(R2)	0000	11	11	1	0	JCC 23	11	START OF O/P AREA IN R3	
019	23	07	01		F2 R1	SDR(R3)	7777	11	11	1	0	JCC 24	11		
020	24	07	01		F1 R1	LMI(R0)	0000	11	11	1	0	JCC 25	11	POINT TO NEXT SIGMA-e	
														R0 to MAR	
														1 + R0 to R0	
021	25	07	01		F4 R1	CLR(R1)	0000	00	11	1	0	JCC 26	00	0 to R1	READ
022	26	07	01		F5 R2	LTM(AC)	7777	00	11	1	0	JCC 27	00	FETCH SIGMA-e	READ
														(CODE) M to AC	
023	27	07	01		F1 R2	LMM(T)	0500	00	11	1	0	JFL 16	00	CONVERT SIGMA-e	READ
														TO POWER. 50M to MAR	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
024	16	03	01	RPG04	F2 R1	SDR(R4)	7777	11	11	1	0	JCR 00	00	SIGMA-e to R4 CODE	READ
025	16	00	01		F5 R2	LTM(AC)	7777	00	11	1	0	JCC 17	00	SIGMA-e (POWER) M to AC	READ
026	17	00	01		F1 R1	LMI(R1)	0603	00	11	1	0	JCC 15	11	ADDRESS OF T'1 603 to MAR	
027	15	00	01		F2 R1	SDR(R5)	7777	11	11	1	0	JCC 14	11	SAVE SIGMA-e (POWER) IN R5	
028	14	00	01		F0 R1	ILR(R4)	0000	00	11	1	0	JCC 13	11	RETURN SIGMA-e (CODE) TO AC R4 to AC	
029	13	00	01	RPG05	F1 R1	LMI(R3)	0000	11	11	1	0	JCR 07	11	POINT TO O/P AREA R R3 to MAR 1 + R3 to R3	
030	13	07	01		F3 R1	INR(R7)	0000	11	10	1	0	JCC 14	00	PREPARE FOR INNER LOOP TEST 1 + R7 to R7	READ
031	14	07	01		F7 R3	XNI(AC)	7777	00	11	1	0	JCC 15	00	XOR SIGMA-e OR SIGMA-exT' TO M	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
032	15	07	01		F1 R1	LMI (R1)	0000	11	11	1	0	JCC 18	10	R1 to MAR	WRITE
														1 + R1 to MAR	
033	18	07	01		F0 R1	ILR(R5)	0000	00	11	1	0	JCF 19	00	RESTORE SIGMA-e	READ
														(POWER)	
														R5 to AC JMP (RPG06, RPG01)	
034	19	02	01	RPG06	F5 R2	LTM(T)	7777	00	11	1	0	JCR 00	00	h = T' to T	READ
														TMSYP TERM to T	
035	19	00	01		F0 R2	AMA(AC)	7777	00	11	1	0	JFL 20	00	SIGMA-e x T'	READ
														(POWER)	
														ZERO TEST	
036	20	02	01	RPG07	F3 R1	INR(R7)	0000	11	10	1	0	JCR 07	11	CHECK FOR END	
														OF INNER LOOP	
														M + AC to AC	
														1 + R7 to R7	
037	20	07	01		F1 R1	LMI(R1)	0000	11	11	1	0	JCC 21	11	POINT TO NEXT T'	
														R1 to MAR	
														1 + R1 to MAR	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION
038	21	07	01		F3 R1	INR(R3)	0000	11	11	1	0	JCF 19	00	INC O/P POINTER READ
														1 + R3 to R3
039	20	03	01	RPG08	F1 R1	LMI(AC)	0400	00	11	1	0	JCR 00	11	POINT TO SIGMA-e x T'
														(CODE)
040	20	00	01		F6 R1	NOP(R0)	0000	00	11	1	0	JCC 18	00	CONVERT TO CODE READ
041	18	00	01		F5 R2	LTM(AC)	7777	00	11	1	0	JCC 13	00	M to AC = READ
														CODE FORM OF RESULT
042	19	03	01		F3 R1	INR(R8)	0000	11	11	1	0	JCR 07	11	1 + R8 to R8
043	19	07	01		F0 R1	ILR(R6)	0000	11	11	1	0	JFL 17	11	$\overline{s} + t + 1$ to R7

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* SOURCE PROGRAM RPVAL */												
	/* READ NUMBER OF ERASURES FOR CURRENT BLOCK */												
	/* STORE ONE'S COMPLEMENT OF NUMBER OF ERASURES IN RO*/												
1	00 13 01	RPV00	F0 R1	ILR	0000	00	00	1	0	JCC 01	11	R9 to AC	
2	01 13 01		FI RI	LMI	0070	00	00	1	0	JCC 02	11	70 Ø TO AC OR 70 Ø + 200 TO AC	
3	02 13 01		FI RI	LMI	0000	11	00	1	0	JCC 03	11	AC to MAR, 1 + AC to AC	
4	03 13 01		F2 RI	SDR	7777	11	00	1	0	JCC 04	00	AC to R6	READ
5	04 13 01		F5 R2	LTM	7777	00	00	1	0	JCC 05	00	M to T	READ
6	05 13 01		F2 R3	LDI	7777	11	00	1	0	JCC 06	00	1 = $\overline{M}$ to AC	READ
7	06 13 01		F2 RI	SDR	7777	11	00	1	0	JCC 07	11	AC to RO	
	/* READ NUMBER OF ERRORS */												
	/* STORE ONE'S COMPLLMENT OF NUMBER OF ERRORS IN R, */												
	/* COMPUTE NUMBER OF ERRATA (ERRORS + ERASURES) STORE ONE'S COMPLEMENT IN R4 */												
8	07 13 01		F4 R1	CLR	0000	00	00	1	0	JCC 08	11	0 to R7	
9	08 13 01		FI RI	LMI	0646	11	00	1	0	JCC 09	11	646 Ø to MAR 647 Ø to R7	
10	09 13 01		F0 RI	ILR	0000	00	00	1	0	JCC 10	00	T to AC	READ
11	10 13 01		F0 R2	AMA	7777	00	00	1	0	JCC 11	00	M + AC to T	READ
12	11 13 01		F2 R3	LDI	7777	11	00	1	0	JCC 12	00	I = $\overline{M}$ to AC	READ

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
13	12 13 01		F2 R1	SDR	7777	11	00	1	0	JCC 13	11	AC to R1	
14	13 13 01		FI R3	CIA	0000	00	00	1	0	JCC 14	11	$\overline{T}$ to T	
15	14 13 01		FO RI	ILR	0000	00	00	1	0	JCC 15	11	T to AC	
16	15 13 01		F2 RI	SDR	7777	11	00	1	0	JCC 16	11	AC to R4	
	/* CLEAR R2 AND R 3 to ZERO */												
	/* INITIALIZE INDEXES FOR TRVOB, TRVOA */												
17	16 13 01		F4 RI	CLR	0000	00	00	1	0	JCC 17	11	0 to R2	
18	17 13 01		F4 RI	CLR	0000	00	00	1	0	JCC 18	11	0 to R3	
19	18 13 01		F2 R1	CSR	0000	00	00	1	0	JCC 19	11	7777 TO AC	
20	19 13 01		F2 RI	CSR	0131	11	00	1	0	JCC 20	11	131 Ø to R8	
21	20 13 01		F2 RI	CSR	0152	11	00	1	0	JCC 21	11	152 Ø to R5	
	/* READ NEXT ERASURE *1												
	/* INCREMENT ONE'S COMPLEMENT OF NUMBER OF ERASURES */												
	/* CHECK IF ALL ERASURES ARE PROCESSED */												
22	21 13 01		FI RI	LMI	0000	11	00	1	0	JCC 22	11	R6 to MAR, 1 + R6 to R6	
23	22 13 01	RPV07	FO RI	ILR	0000	11	00	1	0	JCC 23	00	1 + R0 to R0	READ
24	23 13 01		F2 R3	LDI	7777	11	00	1	0	JFL 24	00	I = $\overline{M}$ to AC	READ

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* ERASURES NOT PROCESSED */												
	/* CHECK IF ERASURE POSITION NUMBER IS ≥ 16 */												
	/* CHECK IF ERASURE POSITION = ZERO */												
25	24 10 01	RPV01	F5 R2	LTM	7777	00	00	1	0	JCR 13	00	M to T	READ
26	24 13 01		F1 R1	LMI	0020	00	00	1	0	JCC 26	11	AC + 20 Ø to AC	
27	26 13 01		F5 R3	TZA	7777	00	00	1	0	JFL 26	11	0 ∨ T to CO	
	/* ERASURES , 16 */												
	/* WRITE ERASURE IN TRVOB INCREMENT NO OF E , 16 */												
28	26 10 01	RPV43	F0 R1	ILR	0000	00	00	1	0	JCC 25	11	T to AC	
29	25 10 01		F1 R1	LMI	0000	11	00	1	0	JCR 12	11	R8 to MAR. 1 + R8 to R8	
30	25 12 01		F3 R1	INR	0000	11	00	1	0	JCR 11	10	1 + R2 to R2	WRITE
	/* ERASURE POSITION ≱ 16 */												
	/* CHECK ERASURE POSITION FOR ZERO VALUE */												
31	26 11 01	RPV03	F6 R1	NOP	0000	00	00	1	0	JFL 28	11		
	/* ERASURE ≠ 0 */												
	/* READ CODE VALUE OF CURRENT ERASURE WRITE IN TRVOA */												
32	28 11 01	RPV04	F0 R1	ILR	0000	00	00	1	0	JCC 25	11	T to AC	
33	25 11 01		F1 R1	LMI	0400	00	00	1	0	JCR 13	11	400 ∨ AC to MAR	
34	25 13 01		F6 R1	NOP	0000	00	00	1	0	JCC 27	00	R0	READ

RECORD NUMBER	ADDRESS R C P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
35	27 13 01		F5 R2	LTM	7777	00	00	1	0	JCC 28	00	M to AC	READ
36	28 13 01	RPV05	FI RI	LMI	0000	11	00	1	0	JCC 29	11	R5 to MAR 1 + R5 to R5	
37	29 13 01		FI RI	LMI	0000	11	00	1	0	JCC 22	10	R6 to MAR 1 + R6 to R6	WRITE
	/* ERASURE POSITION IS ZERO - INSERT CODE 0 0 1 IN ACCUMULATOR */												
38	28 10 01	RPV06	F0 R2	ACM	0000	11	00	1	0	JCR 13	11	1 to AC	
	/* ALL ERASURES PROCESSED READ NEXT ERROR */												
	/* INCREMENT ONE'S COMPLEMENT OF NUMBER OF ERROR */												
	/* CHECK IF ALL ERROS ARE PROCESSED */												
39	24 11 01	RPV08	FI RI	LMI	0000	11	00	1	0	JCR 12	11	R7 TO MAR 1 + 7 TO R7	
40	24 12 01	RPV14	F0 RI	ILR	0000	11	00	1	0	JCC 26	00	1 + R1 TO R1	READ
41	26 12 01		F2 R3	LDI	7777	11	00	1	0	JFL 27	00	1 = $\bar{M}$ to AC	READ
	/* ERRORS NOT ALL PROCESSED */												
	/* CHECK IF ERROR POSITION NUMBER IS , 16 */												
42	27 10 01	RPV09	F5 R2	LTM	7777	00	00	1	0	JCR 12	00	M to T	READ
43	27 12 01		FI RI	LMI	0020	00	00	1	0	JCC 03	11	AC + 2Ø to AC	
44	3 12 01		F5 R3	TZA	7777	00	00	1	0	JFL 12	11	OVT to CO	
	/* ERROR ⩾ 16 */												
	/* WRITE ERROR IN TRVOB, INCREMENT No OF ERROS ⩾ 16*/												

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* ERROR ≥ 16 */														
	/* WRITE ERROR IN TRVOB, INCREMENT NO. OF ERRORS ≥ 16 */														
45	12	10	01	RPV10	F0 R1	ILR	0000	00	00	1	0	JCC 30	11	T to AC	
46	30	10	01		F1 R1	LMI	0000	11	00	1	0	JCR 12	11	R8 to MAR	
														1 + R8 to R8	
47	30	12	01		F3 R1	INR	0000	11	00	1	0	JCR 11	10	1 + R3 to R3	WRITE
	/* ERROR ≱ 16 CHECK ERROR POSITION FOR ZERO VALUE */														
48	12	11	01	RPV11	F6 R1	NOP	0000	00	00	1	0	JFL 9	11		
	ERROR ≠ 0														
	/* READ CODE VALUE OF CURRENT ERROR. WRITE IN TRVOA */														
49	09	11		RPV12	F0 R1	ILR	0000	00	00	1	0	JCC 30	11	T to A	
50	30	11	01		F1 R1	LMI	0400	00	00	1	0	JCR 13	11	400_V AC to MAR	
51	30	13	01		F6 R1	NOP	0000	00	00	1	0	JCC 31	00	R0	READ
52	31	13	01		F5 R2	LTM	7777	00	00	1	0	JCR 12	00	M to AC	READ
	/* WRITE CODE VALUE IN TRVOA */														
53	31	12	01		F1 R1	LMI	0000	11	00	1	0	JCC 29	11	R5 to MAR	
														1 + R5 to R5	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
54	29	12	01	RPV13	F1 R1	LMI	0000	11	00	1	0	JCC 24	10	R7 to MAR	WRITE
														1 + R7 to R7	

/* ERROR POSITION = ZERO */

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION
55	09	10	01	RPV15	F0 R2	ACM	0000	11	00	1	0	JCR 12	11	1 to AC
56	09	12	01		F1 R1	LMI	0000	11	00	1	0	JCC 29	11	R5 to MAR
														1 + R5 to R5

/* INITIALIZE INDEXES FOR OUTPUT TABLE TRVOC */

/* INITIALIZE INDEX FOR INPUT TABLE TRVOB */

/* COMPUTE NUMBER OF ERRATA WITH POSITIONS ⩾16, WRITE IN TABLE TRVOC */

/* STORE ONE'S COMPLEMENT OF NO. OF ERRATA ⩾16 in R_0, R_1*/

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION
57	27	11	01	RPV16	F4 R1	CLR	0000	00	00	1	0	JCR 14	11	0 to R7
58	27	14	01		F4 R1	CLR	0000	00	00	1	0	JCC 23	11	0 to R8
59	23	14	01		F0 R1	ILR	0000	00	00	1	0	JCR 12	11	R2 to AC
60	23	12	01		F0 R1	ALR	7777	00	00	1	0	JCC 22	11	R3 + AC to AC
61	22	12	01		F1 R1	LMI	0330	11	00	1	0	JCC 21	11	330Ø to MAR
														331Ø to R7

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
62	21	12	01		F1 R3	CIA	0000	00	00	1	0	JCC 20	10	$\overline{AC}$ to AC	WRITE
63	20	12	01		F2 R1	SDR	7777	11	00	1	0	JCC 19	11	AC to R0	
64	19	12	01		F2 R1	SDR	7777	11	00	1	0	JCC 18	11	AC to R1	
65	18	12	01		F1 R1	LMI	0131	11	00	1	0	JCC 17	11	131Ø to MAR 132Ø to R8	
/* READ NEXT ERRATA WITH POSITION ≥ FROM TRVOB */															
/* INCREMENT THE NO. OF ERRATA ≥ 16 */															
/* CHECK IF ALL ERRATA ≥ 16 ARE PROCESSED */															
66	17	12	01	RPV21	F3 R1	INR	0000	11	00	1	0	JCC 16	00	1 + R0 to R0	READ
67	16	12	01		F1 R2	LMM	0400	00	00	1	0	JFL 31	00	K_VM to MAR 400Ø + M to AC	READ
/* ERRATA NOT YET ALL PROCESSED */															
/* STORE CURRENT ERRATA IN 77 R2 - POWER FORM */															
/* READ CODE VALUE OF CURRENT ERRATA. STORE IN R6 */															
68	31	10	01	RPV17	F2 R1	CSR	0037	11	00	1	0	JCR 14	00	AC to R2	READ
69	31	14	01		F5 R2	LTM	7777	00	00	1	0	JCC 30	00	M to AC	READ
70	30	14	01		F2 R1	SDR	7777	11	00	1	0	JCC 29	11	AC to R6	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* INITIALIZE INDEX FOR TABLE OF ALL ERRATA TRVOA */														
	/* STORE NUMBER OF ALL ERRATA IN R3 */														
71	29	14	01		F2 R1	CSR	0000	00	00	1	0	JCC 28	11	All 1's to AC	
72	28	14	01		F2 R1	CSR	0152	11	00	1	0	JCC 26	11	152Ø to R5	
73	26	14	01		F0 R1	ILR	0000	00	00	1	0	JCC 25	11	R4 to AC	
74	25	14	01		F2 R1	SDR	7777	11	00	1	0	JCC 24	11	AC to R3	
	/* READ NEXT ERRATA FROM TRVOA (CODE) */														
	/* INCREMENT ONE'S COMPLEMENT OF NO. OF ERRATA IN TRVOA */														
	/* CHECK IF ALL ERRATA PROCESSED */														
75	24	14	01	RPV20	F0 R1	ILR	0000	00	00	1	0	JCC 22	11	R6 to AC	
76	22	14	01		F1 R1	LMI	0000	11	00	1	0	JCC 21	11	R5 to MAR	
														1 + R5 to R5	
77	21	14	01		F3 R1	INR	0000	11	00	1	0	JCC 20	00	1 + R3 to R3	READ
78	20	14	01		F7 R3	XNI	7777	00	00	1	0	JFL 19	00	I = $\overline{M} \oplus$ AC to AC	READ
	/* ERRATA NOT ALL PROCESSED */														
	/* XOR CURRENT ERRATA WITH CODE IN R6 */														
	/* CONVERT THE RESULT TO POWER */														

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* CUMULATIVELY ADD INTO π IN R2 ADJUST FOR POWER INDEX */														
79	19	10	01	RPV18	F1 R1	LMI	0500	00	00	1	0	JCR 15	11	AC to MAR	
80	19	15	01		F0 R1	ILR	0000	00	00	1	0	JCC 20	00	R2 to AC	READ
81	20	15	01		F0 R2	AMA	7777	00	00	1	0	JCC 18	00	M + AC to AC	READ
82	18	15	01		F3 R3	AIA	7777	00	00	1	0	JCR 14	11	AC + I to AC	
83	18	14	01		F2 R1	CSR	0037	11	00	1	0	JCC 24	11	$\overset{K}{\text{AC}}$ to R2	
	/* ERRATA IN TABLE TRVOA ALL PROCESSED */														
	/* WRITE CURRENT PRODUCT IN TRVOC */														
84	19	11	01	RPV19	F0 R1	ILR	0000	00	00	1	0	JCR 14	11	R2 to AC	
85	19	14	01		F1 R1	LMI	0000	11	00	1	0	JCC 17	11	R7 to MAR	
														1 + R7 to R7	
86	17	14	01		F1 R1	LMI	0000	11	00	1	0	JCR 12	10	R8 to MAR	WRITE
														1 + R8 to R8	
	/* ALL ERRATA IN TRVOB PROCESSED */														
	/* EVALUATE Z POLYNOMIAL */														
	/* INITIALIZE INDEXES FOR TABLES TRVOB, TRVOC, TMSYN */														
	/* INITIALIZE INDEX FOR CHARACTER POSITIONS IN TADIØ/TBDIØ */														

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
87	31	11	01	RPV22	F2 R1	CSR	0000	00	00	1	0	JCR 15	11	All 1's to AC	
88	31	15	01		F2 R1	CSR	0131	11	00	1	0	JCC 30	11	131 Ø to R8	
89	30	15	01		F2 R1	CSR	0331	11	00	1	0	JCC 29	11	331 Ø to R7	
90	29	15	01		F2 R1	CSR	0562	11	00	1	0	JCC 24	11	562 Ø to R4	
91	24	15	01		F1 R1	LMI	0007	00	00	1	0	JCC 28	11	R9 + 7 to R9	

/* READ NUMBER OF COEFFICIENTS IN ERRATA POLYNOMIAL */

/* STORE THE TWO'S COMPLEMENT OF NUMBER OF COEFFICIENTS IN R6 */

/* SET WRITE EARLY FLAG R2 TO ZERO */

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
92	28	15	01		F1 R2	LMM	0561	00	00	1	0	JCC 27	11	561 Ø to MAR 561 Ø to AC	
93	27	15	01		F4 R1	CLR	0000	00	00	1	0	JCC 26	00	0 to AC	READ
94	26	15	01		F3 R3	AIA	7777	11	00	1	0	JCC 25	00	$1 + I(\overline{M})$ to AC	READ
95	25	15	01		F2 R1	SDR	7777	11	00	1	0	JCC 12	11	AC to R6	
96	12	15	01		F4 R1	CLR	0000	00	00	1	0	JCC 21	11	0 to R2	

97 /* READ NEXT ERRATA FROM TRVOB */

/* INCREMENT ONE'S COMPLEMENT OF NO. OF ERRATA ⩾16 */

/* CHECK IF ALL ERRATA HAVE BEEN PROCESSED */

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
98	21	15	01	RPV39	F1 R1	LMI	0000	11	00	1	0	JCC 17	11	R8 to MAR	
														1 + R8 to R8	
99	17	15	01		F3 R1	INR	0000	11	00	1	0	JCC 16	00	1 + R1 to R1	READ
100	16	15	01		F5 R2	LTM	7777	00	00	1	0	JFL 21	00	M to A	READ
	/* ALL ERRATA NOT PROCESSED */														
	/* CHECK IF 2 OR LESS ERRATA REMAIN */														
101	21	10	01		F1 R1	LMI	0002	00	00	1	CPE	JCR 9		R1 + 2 to R1	
102	21	09	01		F5 R1	TZR	7777	00	00	1	0	JFL 20		O_V R2	
	/* TWO OR LESS ERRATA REMAIN */														
	/* CHECK WRITE EARLY FLAG R2 */														
103	20	11	01	RPV24	F2 R1	SDR	7777	11	00	1	0	JFL 18		AC to R3	
	/* WRITE EARLY FLAG NOT SET. SET R2 $\neq$ 0 */														
	/* COMPUTE ADDRESS OF WRITE EARLY FLAG FOR NON CURRENT BLOCK */														
	/* WRITE INTO RAM */														
104	18	10	01	RPV25	F0 R1	ILR	0000	00	00	1	0	JCR 9		R9 to AC	
105	18	09	01		F1 R3	CIA	0000	00	00	1	0	JCC 19		$\overline{AC}$ to AC	
106	19	09	01		F5 R1	CLA	0200	00	00	1	0	JCC 20		$K_\wedge$AC to AC (0 or 200 to AC)	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
107	20	09	01		F1 R1	LMI	0177	00	00	1	0	JCC 25	11	AC + 177 Ø to AC to MAR	
108	25	09	01		F1 R3	CIA	0000	11	00	1	0	JCC 23	11	$\overline{AC}$ + 1 to AC (001)	
109	23	09	01		F2 R1	SDR	7777	11	00	1	0	JCC 24	10	AC to R2	WRITE
110	24	09	01		F0 R1	ILR	0000	00	00	1	0	JCC 22	11	R3 to AC	
111	22	09	01		F1 R1	LMI	0400	00	00	1	0	JCR 15	11	400 $\vee$ AC to MAR	
	/* WRITE EARLY FLAG SET */														
112	18	11	01	RPV26	F6 R1	NOP	0000	00	00	1	0	JCC 22	11		
113	22	11	01		F1 R1	LMI	0400	00	00	1	0	JCR 15	11	400 $\vee$ AC to MAR	
	/* MORE THAN TWO ERRATA LEFT */														
114	20	10	01	RPV27	F2 R1	SDR	7777	11	00	1	0	JCC 22	11	AC to R3	
	/* ALL ERRATA NOT PROCESSED */														
	/* STORE CURRENT ERRATA IN R3 */														
	/* CONVERT CURRENT ERRATA TO CODE AND STORE IN T */														
	/* INITIALIZE ONE'S COMPLEMENT OF NO. OF COEFFICIENTS IN ERRATA POLYNOMIAL */														
	/* INITIALIZE TMSYN INDEX */														
115	22	10	01	RPV28	F1 R1	LMI	0400	00	00	1	0	JCR 15	11	400 $\vee$ AC to MAR	
116	22	15	01	RPV29	F0 R1	ILR	0000	00	00	1	0	JCC 10	00	R4 to AC	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
117	10	15	01		F5 R2	LTM	7777	00	00	1	0	JCC 23	00	M to T	READ
118	23	15	01		F2 R1	SDR	7777	11	00	1	0	JCC 14	11	AC to R0	
119	14	15	01		F0 R1	ILR	0000	00	00	1	0	JCC 13	11	R6 to AC	
120	13	15	01		F2 R1	SDR	7777	11	00	1	0	JCC 11	11	AC to R5	
	/* READ NEXT ERRATA COEFFICIENT FROM TMSYN */														
	/* INCREMENT THE ONE'S COMPLEMENT OF NUMBER OF COEFFICIENTS */														
	/* CHECK IF ALL COEFFICIENTS LESS ONE ARE PROCESSED */														
121	11	15	01	RPV30	F1 R1	LMI	0000	11	00	1	0	JCC 09	11	1 + R0 to R0 R0 to MAR	
122	09	15	01		F3 R1	INR	0000	11	00	1	0	JCC 04	00	1 + R5 to R5	READ
123	04	15	01		F6 R1	NOP	0000	00	00	1	0	JFL 15	00	0 to R0	READ
	/* ALL COEFFICIENTS NOT PROCESSED */														
	/* XOR COEFFICIENT WITH SUM IN T AND CHECK RESULT FOR ZERO */														
124	15	10	01	RPV31	F7 R3	XNI	7777	00	00	1	0	JCR 14	00	$I \overline{\oplus} T$ to T	READ
125	15	14	01		F5 R3	TZA	7777	00	00	1	0	JCC 05	11	$O \vee T$ to CO	
126	05	14	01		F1 R1	LMI	0500	00	00	1	0	JFL 11	11	$500 \vee T$ to MAR 500 + T to T	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* RESULT NOT ZERO */														
	/* CONVERT RESULT TO POWER AND ADD B_i TERM IN R3 */														
	/* CONVERT RESULT TO CODE AND STORE IN CUMULATIVE SUM T */														
127	11	11	01	RPV32	F0 R1	ILR	0000	00	00	1	0	JCR 12	00	R3 to AC	READ
128	11	12	01		F0 R2	AMA	7777	00	00	1	0	JCC 08	00	M + AC to AC	READ
129	08	12	01		F1 R1	LMI	0400	00	00	1	0	JCR 15	11	400_V AC to MAR	
														400 + AC to AC	
130	08	15	01		F6 R1	NOP	0000	00	00	1	0	JCC 07	00	0 to R0	READ
131	07	15	01		F5 R2	LTM	7777	00	00	1	0	JCC 02	00	M to AC	READ
132	02	15	01		F2 R1	SDR	7777	11	00	1	0	JCC 11	11	A to T	
	/* RESULT IS ZERO. SET SUM T = ZERO */														
133	11	10	01	RPV33	F4 R1	CLR	0000	00	00	1	0	JCR 15	11	0 to T	
	/* ALL COEFFICIENTS LESS ONE PROCESSED */														
	/* READ LAST COEFFICIENT AND XOR INTO SUM IN T */														
	/* CHECK IF RESULT IS ZERO (NUMERATOR) */														
	/* CONVERT THE RESULT TO POWER */														
134	15	11	01	RPV34	F1 R1	DSM	7777	00	00	1	0	JCR 15	11	All 1's to MAR	
														R0 - 1 to R0	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
135	00	15	01		F1 R1	LMI	0000	00	00	1	0	JCC 15	11	1 + R0 to R0	
														R0 to MAR	
136	15	15	01		F6 R1	NOP	0000	00	00	1	0	JCR 12	00	0 to R0	READ
137	15	12	01		F7 R3	XNI	7777	00	00	1	0	JCC 05	00	$\overline{I} \oplus T$ to T	READ
138	05	12	01		F5 R3	TZA	7777	00	00	1	0	JCC 04	11	O_V T to CO	
139	04	12	01		F6 R1	NOP	0000	00	00	1	0	JFL 13	11	0 to R0	
	/* RESULT $\neq$ ZERO */														
140	13	11	01	RPV35	F1 R1	LMI	0500	00	00	1	0	JCR 12	11	500 $_V$ T to MAR	
141	13	12	01		F6 R1	NOP	0000	00	00	1	0	JCC 12	00	0 to R0	READ
142	12	12	01		F5 R2	LTM	7777	00	00	1	0	JCC 14	00	M to AC	READ
	/* SUM IN T NOT ZERO */														
	/* READ NEXT PRODUCT TERM FROM TRVOC */														
	/* COMPUTE ERROR CORRECTION TERM Z $(B_i)/B_i \prod (B_i + B_j)$ */														
	/* CHECK ERROR CORRECTION TERM FOR -ve POWER */														
143	14	12	01		F1 R1	LMI	0000	11	00	1	0	JCC 10	11	R7 to MAR	
														1 + R7 to R7	
144	10	12	01		F6 R1	NOP	0000	00	00	1	0	JCC 07	00	O to R0	READ

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
145	07	12	01		F3 R3	AIA	7777	00	00	1	0	JCC 06	00	AC + I ($\overline{M}$) to AC	READ
146	06	12	01		F1 R2	LMM	0037	00	00	1	0	JFL 14	11	37 Ø to T	
														37 Ø to MAR	
	/* NEGATIVE RESULT OR ZERO RESULT */														
	/* CONVERT TO POSITIVE POWER BY ADDING 31D (37 Ø) */														
147	14	10	01	RPV36	F0 R1	ALR	7777	00	00	1	0	JCR 11	11	T + AC to T, AC	
	/* POSITIVE RESULT */														
	/* CONVERT RESULT TO CODE. STORE THE TWO'S COMPLEMENT IN T */														
148	14	11	01	RPV37	F1 R1	LMI	0000	11	00	1	0	JCR 08	11	1 + AC to AC	
149	14	08	01		F1 R1	LMI	0400	00	00	1	0	JCR 14	11	400 $_V$ AC to MAR	
150	14	14	01		F0 R1	ILR	0000	00	00	1	0	JCC 13	00	R3 to AC	READ
151	13	14	01		F5 R2	LTM	7777	00	00	1	0	JCC 10	00	M to T	READ
	/* SUM IN T IS ZERO. */														
	/* COMPUTE ADDRESS OF CHARACTER CORRESPONDING TO ERRATA POSITION IN R3 */														
	/* READ CHARACTER FROM TADIØ/TBDIØ */														
152	10	14	01	RPV40	F1 R3	CIA	0000	00	00	1	0	JCC 11	11	$\overline{AC}$ to AC	
153	11	14	01		F2 R1	CSR	0037	11	00	1	0	JCC 08	11	AC to R3	

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
154	08	14	01		F0 R1	ILR	0000	00	00	1	0	JCC 07	11	R9 to AC	
155	07	14	01		F0 R1	ALR	7777	00	00	1	0	JCC 06	11	AC + R3 to R3, AC	
	/* XOR CHARACTER WITH ERROR CORRECTION TERM IN T */														
156	06	14	01		F1 R1	LMI	0000	00	00	1	0	JCC 12	11	$O \vee$ R3 to MAR	
157	12	14	01		F0 R1	ILR	0000	00	00	1	0	JCC 09	00	T to AC	READ
158	09	14	01		F7 R3	XNI	7777	00	00	1	0	JCC 03	00	$I(\overline{M}) \overline{\oplus}$ AC to AC	READ
	/* WRITE CORRECTED CHARACTER IN TADIØ / TBDIØ */														
159	03	14	01		F1 R1	LMI	0000	00	00	1	0	JCR 15	11	$0 \vee$ R3 to MAR	
160	03	15	01		F1 R1	LMI	0000	11	00	1	0	JCC 17	10	R8 to MAR 1 + R8 to R8	WRITE
	/* SUM IN T IS ZERO */														
161	13	10	01	RPV38	F0 R1	ILR	0000	00	00	1	0	JCC 10	11	R3 to AC	
162	10	10	01		F3 R1	INR	0000	11	00	1	0	JCR 14	11	1 + R7 to R7	
	/* ALL ERRATA $\geqslant$16 PROCESSED */														
	/* RESTORE R9 (0 OR 200 Ø) */														
163	21	11	01	RPV41	F5 R1	LRI	0700	00	00	1	0	JCC 10	11	K $\wedge$ R9 to R9	
	/* READ ADDRESS OF ERROR POLYNOMIAL */														

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
	/* READ NUMBER OF ERRORS AND WRITE IN DATA QUALITY WORD */														
164	10	11	01		F1 R2	LMM	0644	00	00	1	0	JCR 09	11	644 Ø to MAR	
														644 Ø to AC	
165	10	09	01		F0 R1	ILR	0000	00	00	1	0	JCC 11	00	R9 to AC	READ
166	11	09	01		F1 R2	LMM	0700	00	00	1	0	JCC 12	00	0_V M to MAR	READ
														M to T	
167	12	09	01		F2 R1	SDR	7777	11	00	1	0	JCC 13	00	AC to T	READ
168	13	09	01		F0 R2	ACM	0000	00	00	1	0	JCC 14	00	M to AC	READ
169	14	09	01		F1 R1	LMI	0007	00	00	1	0	JCC 17	11	7 + T to T	
														7_V T to MAR	
170	17	09	01		F5 R1	TZR	7777	00	00	1	0	JCC 15	10	0_V R2 to CO	WRITE
	/* CHECK IF WRITE EARLY FLAG IS SET */														
171	15	09	01		F0 R1	ILR	0000	00	00	1	0	JFL 04	11	R9 to AC	
	/* WRITE EARLY FLAG IS SET */														
172	04	11	01	RPV42	F6 R1	NOP	0000	00	00	1	0	JCR 14	11		
173	04	14	01		F5 R1	TZR	7777	00	00	1	0	JCC 16	11	0_V R9 to CO	
	/* EXIT TO INOUT AT IOC03 */														
	/* WRITE EARLY FLAG IS NOT SET */														

/* COMPUTE ADDRESS OF WRITE EARLY FLAG FOR NON-CURRENT BLOCK */

/* SET WRITE REQUEST IN RAM */

RECORD NUMBER	ADDRESS R	C	P	LABEL	CPE	MNEM	KBUS	CI	CO	LOAD	INH	JUMP	READ/ WRITE	MICROFUNCTION	
174	04	10	01	RPV02	F1 R3	CIA	0000	00	00	1	0	JCR 09	11	$\overline{AC}$ to AC	
175	04	09	01		F5 R1	LMI	0200	00	00	1	0	JCC 03	11	K $\wedge$ AC to AC	
176	03	09	01		F1 R1	LMI	0177	00	00	1	0	JCC 02	11	AC + 177Ø to AC to MAR	
177	02	09	01		F1 R3	CIA	0000	11	00	1	0	JCR 14	11	$\overline{AC}$ + 1 to AC (001)	
178	02	14	01		F5 R1	TZR	7777	00	00	1	0	JCC 16	10	0 $\vee$ R9 to CO	WRITE

/* EXIT TO INOUT AT IOC03 */

INTEL 3000

COMPUTER (MICROPROCESSOR) DATA SHEETS

SCHOTTKY BIPOLAR LSI MICROCOMPUTER SET

3001 MICROPROGRAM CONTROL UNIT

The INTEL® Bipolar Microcomputer Set is a family of Schottky bipolar LSI circuits which simplify the construction of microprogrammed central processors and device controllers. These processors and controllers are truly microprogrammed in the sense that their control logic is organized around a separate read-only memory called the microprogram memory. Control signals for the various processing elements are generated by the microinstructions contained in the microprogram memory. In the implementation of a typical central processor, as shown below, the microprogram interprets a higher level of instructions called macroinstructions, similar to those found in a small computer. For device controllers, the microprograms directly implement the required control functions.

The INTEL® 3001 Microprogram Control Unit (MCU) controls the sequence in which microinstructions are fetched from the microprogram memory. Its functions include the following:

Maintenance of the microprogram address register.

Selection of the next microinstruction based on the contents of the microprogram address register.

Decoding and testing of data supplied via several input busses to determine the microinstruction execution sequence.

Saving and testing of carry output data from the central processor (CP) array.

Control of carry/shift input data to the CP array.

Control of microprogram interrupts.

High Performance — 85 ns Cycle Time

TTL and DTL Compatible

Fully Buffered Three-State and Open Collector Outputs

Direct Addressing of Standard Bipolar PROM or ROM

512 Microinstruction Addressability

Advanced Organization
- **9-Bit Microprogram Address Register and Bus**
- **4-Bit Program Latch**
- **Two Flag Registers**

Eleven Address Control Functions
- **Three Jump and Test Latch Functions**
- **16-way Jump and Test Instruction Bus Function**

Eight Flag Control Functions
- **Four Flag Input Functions**
- **Four Flag Output Functions**

40 Pin DIP

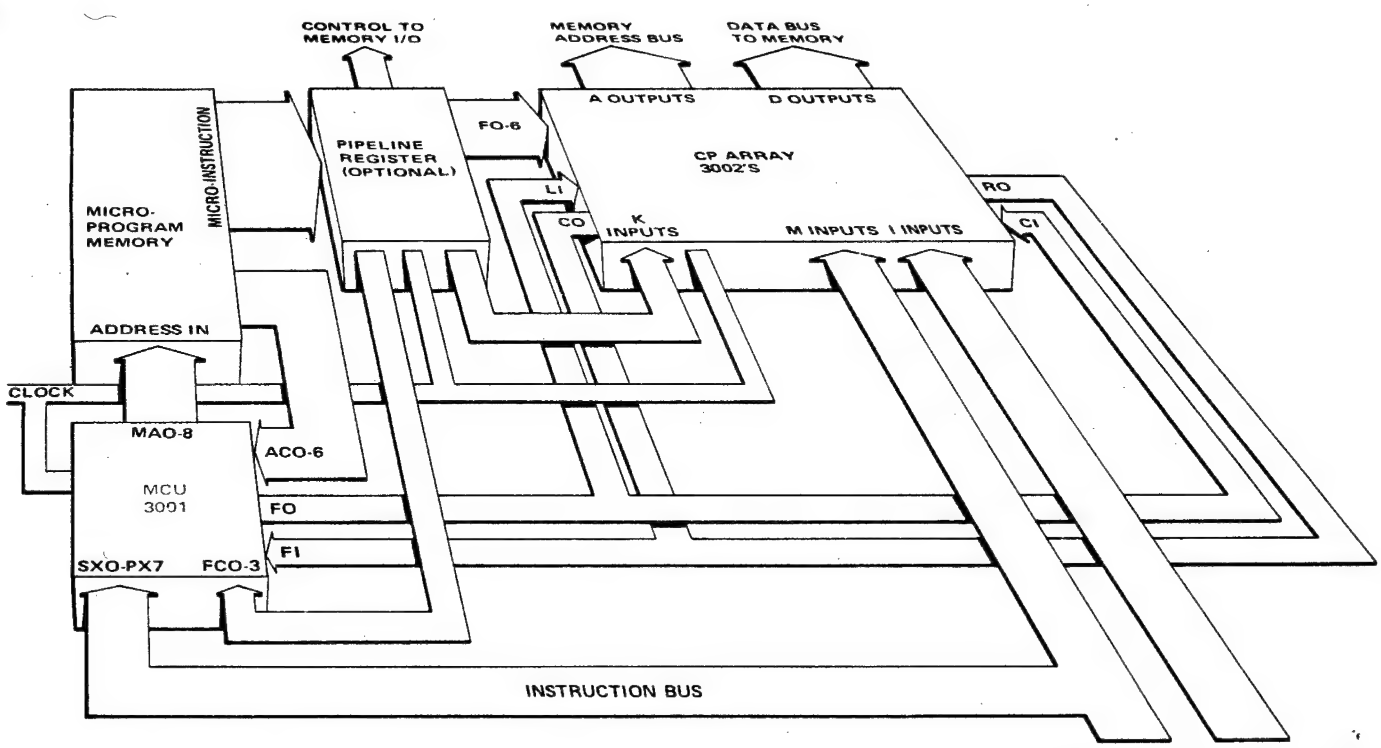


Figure 1. Block Diagram of a Typical System

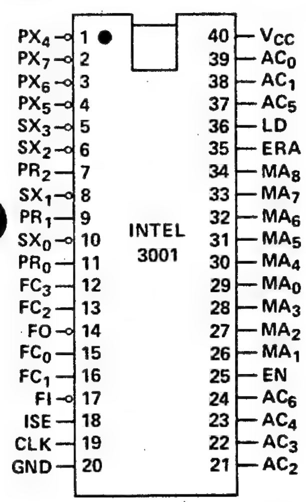
Other members of the INTEL Bipolar Microcomputer Set:

3002	Central Processing Element
3003	Look-Ahead Carry Generator
3212	Multi-Mode Latch Buffer
3214	Priority Interrupt Control Unit
3226	Inverting Bi-Directional Bus Driver
3301	Schottky Bipolar ROM (256 x 4)
3304A	Schottky Bipolar ROM (512 x 8)
3601	Schottky Bipolar PROM (256 x 4)
3604	Schottky Bipolar PROM (512 x 8)

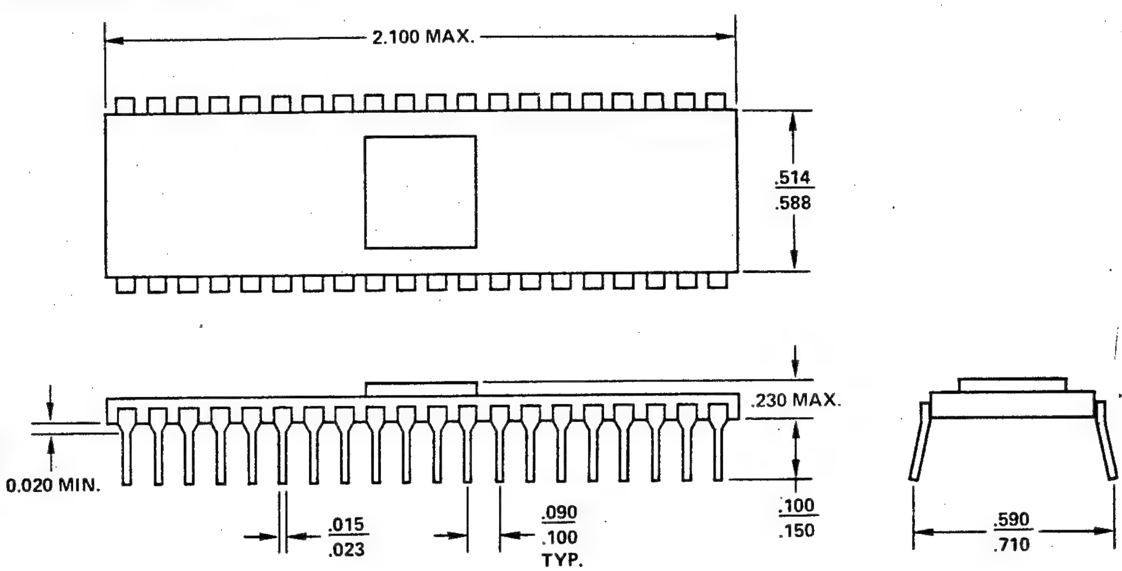
CONTENTS

APPENDICES

PACKAGE CONFIGURATION



PACKAGE OUTLINE



PIN DESCRIPTION

PIN	SYMBOL	NAME AND FUNCTION	TYPE[1]
1-4	PX_4-PX_7	Primary Instruction Bus Inputs Data on the primary instruction bus is tested by the JPX function to determine the next microprogram address.	active LOW
5, 6, 8, 10	SX_0-SX_3	Secondary Instruction Bus Inputs Data on the secondary instruction bus is synchronously loaded into the PR-latch while the data on the PX-bus is being tested (JPX). During a subsequent cycle, the contents of the PR-latch may be tested by the JPR, JLL, or JRL functions to determine the next microprogram address.	active LOW
7, 9, 11	PR_0-PR_2	PR-Latch Outputs The PR-latch outputs are asynchronously enabled by the JCE function. They can be used to modify microinstructions at the outputs of the microprogram memory or to provide additional control lines.	open collector
12, 13, 15, 16	FC_0-FC_3	Flag Logic Control Inputs The flag logic control inputs are used to cross-switch the flags (C and Z) with the flag logic input (FI) and the flag logic output (FO).	
14	FO	Flag Logic Output The outputs of the flags (C and Z) are multiplexed internally to form the common flag logic output. The output may also be forced to a logical 0 or logical 1.	active LOW three-state
17	FI	Flag Logic Input The flag logic input is demultiplexed internally and applied to the inputs of the flags (C and Z). Note: the flag input data is saved in the F-latch when the clock input (CLK) is low.	active LOW
18	ISE	Interrupt Strobe Enable Output The interrupt strobe enable output goes to logical 1 when one of the JZR functions are selected (see Functional Description, page 6). It can be used to provide the strobe signal required by the INTEL 3214 Priority Interrupt Control Unit or other interrupt circuits.	
19	CLK	Clock Input	
20	GND	Ground	
21-24 37-39	AC_0-AC_6	Next Address Control Function Inputs All jump functions are selected by these control lines.	
25	EN	Enable Input When in the HIGH state, the enable input enables the microprogram address, PR-latch and flag outputs.	
26-29	MA_0-MA_3	Microprogram Column Address Outputs	three-state
30-34	MA_4-MA_8	Microprogram Row Address Outputs	three-state
35	ERA	Enable Row Address Input When in the LOW state, the enable row address input independently disables the microprogram row address outputs. It can be used with the INTEL 3214 Priority Interrupt Control Unit or other interrupt circuits to facilitate the implementation of priority interrupt systems.	
36	LD	Microprogram Address Load Input When in the active HIGH state, the microprogram address load input inhibits all jump functions and synchronously loads the data on the instruction busses into the microprogram register. However, it does not inhibit the operation of the PR-latch or the generation of the interrupt strobe enable.	
40	VCC	+5 Volt Supply	

NOTE:
(1) Active HIGH unless otherwise specified.

LOGICAL DESCRIPTION

The MCU performs two major control functions. First, it controls the sequence in which microinstructions are fetched from the microprogram memory. For this purpose, the MCU contains a microprogram address register and the associated logic for selecting the next microinstruction address. The second function of the MCU is the control of the two flag flip-flops that are included for interaction with the carry input and carry output logic of the CP array. The logical organization of the MCU is shown in Figure 2.

NEXT ADDRESS LOGIC

The next address logic of the MCU provides a set of conditional and unconditional address control functions. These address control functions are used to implement a jump or jump/test operation as part of every microinstruction. That is to say, each microinstruction typically contains a jump operation field that specifies the address control function, and hence, the next microprogram address.

In order to minimize the pin count of the MCU, and reduce the complexity of the next address logic, the microprogram address space is organized as a two dimensional array or matrix. Each microprogram address corresponds to a unit of the matrix at a particular row and column location. Thus, the 9-bit microprogram address is treated as specifying not one, but two addresses — the row address in the upper five bits and the column address in the lower four bits. The address matrix can therefore contain, at most, 32 row addresses and 16 column addresses for a total of 512 microinstructions.

The next address logic of the MCU makes extensive use of this two component addressing scheme. For example, from a particular row or column address, it is possible to jump unconditionally in one operation anywhere in that row or column. It is not possible, however, to jump anywhere in the address matrix. In fact, for a given location in the matrix, there is a fixed subset of microprogram addresses that may be selected as the next address. These possible jump target addresses are referred to as a jump set. Each type of MCU address control (jump) function has a jump set associated with it. Appendix C illustrates the jump set for each function.

FLAG LOGIC

The flag logic of the MCU provides a set of functions for saving the current value of the carry output of the CP array and for controlling the value of the carry input to the CP array. These two distinct flag control functions are called flag input functions and flag output functions.

The flag logic is comprised of two flip-flops, designated the C-flag and the Z-flag, along with a simple latch, called the F-latch, that indicates the current state of the carry output line of the CP array. The flag logic is used in conjunction with the carry and shift logic of the CP array to implement a variety of shift/rotate and arithmetic functions.

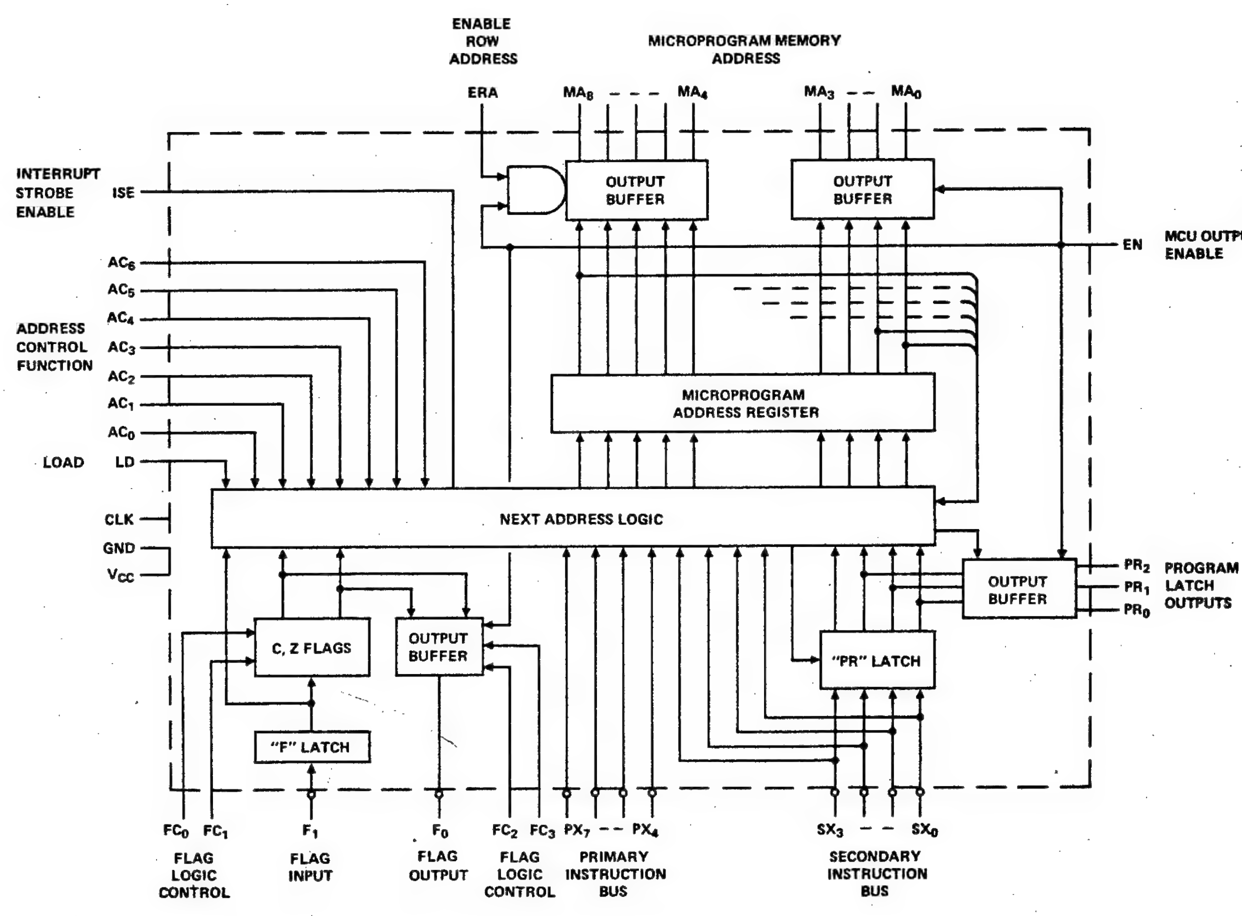


Figure 2. 3001 Block Diagram

ADDRESS CONTROL FUNCTIONS

The address control functions of the MCU are selected by the seven input lines designated AC_0-AC_6. On the rising edge of the clock, the 9-bit microprogram address generated by the next address logic is loaded into the microprogram address register. The next microprogram address is delivered to the microprogram memory via the nine output lines designated MA_0-MA_8. The microprogram address outputs are organized into row and column addresses as:

$$\underbrace{MA_8\ MA_7\ MA_6\ MA_5\ MA_4}_{\text{row address}}$$

$$\underbrace{MA_3\ MA_2\ MA_1\ MA_0}_{\text{column address}}$$

Each address control function is specified by a unique encoding of the data on the function input lines. From three to five bits of the data specify the particular function while the remaining bits are used to select part of either the row or column address desired. Function code formats are given in Appendix A, "Address Control Function Summary."

The following is a detailed description of each of the eleven address control functions. The symbols shown below are used throughout the description to specify row and column addresses.

Symbol	Meaning
row_n	5-bit next row address where n is the decimal row address.
col_n	4-bit next column address where n is the decimal column address.

UNCONDITIONAL ADDRESS CONTROL (JUMP) FUNCTIONS

The jump functions use the current microprogram address (i.e., the contents of the microprogram address register prior to the rising edge of the clock) and several bits from the address control inputs to generate the next microprogram address.

Mnemonic	Function Description
JCC	Jump in current column. AC_0-AC_4 are used to select 1 of 32 row addresses in the current column, specified by MA_0-MA_3, as the next address
JZR	Jump to zero row. AC_0-AC_3 are used to select 1 of 16 column addresses in row_0, as the next address.
JCR	Jump in current row. AC_0-AC_3 are used to select 1 of 16 addresses in the current row, specified by MA_4-MA_8, as the next address.
JCE	Jump in current column/row group and enable PR-latch outputs. AC_0-AC_2 are used to select 1 of 8 row addresses in the current row group, specified by MA_7-MA_8, as the next row address. The current column is specified by MA_0-MA_3. The PR-latch outputs are asynchronously enabled.

FLAG CONDITIONAL ADDRESS CONTROL (JUMP/TEST) FUNCTIONS

The jump/test flag functions use the current microprogram address, the contents of the selected flag or latch, and several bits from the address control function to generate the next microprogram address.

Mnemonic	Function Description
JFL	Jump/test F-Latch. AC_0-AC_3 are used to select 1 of 16 row addresses in the current row group, specified by MA_8, as the next row address. If the current column group, specified by MA_3, is col_0-col_7, the F-latch is used to select col_2 or col_3 as the next column address. If MA_3 specifies column group col_8-col_{15}, the F-latch is used to select col_{10} or col_{11} as the next column address.
JCF	Jump/test C-flag. AC_0-AC_2 are used to select 1 of 8 row addresses in the current row group, specified by MA_7 and MA_8, as the next row address. If the current column group specified by MA_3 is col_0-col_7, the C-flag is used to select col_2 or col_3 as the next column address. If MA_3 specifies column group col_8-col_{15}, the C-flag is used to select col_{10} or col_{11} as the next column address.
JZF	Jump/test Z-flag. Identical to the JCF function described above, except that the Z-flag, rather than the C-flag, is used to select the next column address.

PX-BUS AND PR-LATCH CONDITIONAL ADDRESS CONTROL (JUMP/TEST) FUNCTIONS

The PX-bus jump/test function uses the data on the primary instruction bus (PX_4-PX_7), the current mircoprogram address, and several selection bits from the address control function to generate the next microprogram address. The PR-latch jump/test functions use the data held in the PR-latch, the current microprogram address, and several selection bits from the address control function to generate the next microprogram address.

Mnemonic	Function Description
JPR	Jump/test PR-latch. AC_0-AC_2 are used to select 1 of 8 row addresses in the current row group, specified by MA_7 and MA_8, as the next row address. The four PR-latch bits are used to select 1 of 16 possible column addresses as the next column address.

Mnemonic	Function Description
JLL	Jump/test leftmost PR-latch bits. AC_0-AC_2 are used to select 1 of 8 row addresses in the current row group, specified by MA_7 and MA_8, as the next row address. PR_2 and PR_3 are used to

select 1 of 4 possible column addresses in col_4 through col_7 as the next column address.

JRL Jump/test rightmost PR-latch bits. AC_0 and AC_1 are used to select 1 of 4 high-order row addresses in the current row group, specified by MA_7 and MA_8, as the next row address. PR_0 and PR_1 are used to select 1 of 4 possible column addresses in col_{12} through col_{15} as the next column address.

JPX Jump/test PX-bus and load PR-latch. AC_0 and AC_1 are used to select 1 of 4 row addresses in the current row group, specified by MA_6-MA_8, as the next row address. PX_4-PX_7 are used to select 1 of 16 possible column addresses as the next column address. SX_0-SX_3 data is locked in the PR-latch at the rising edge of the clock.

FLAG CONTROL FUNCTIONS

The flag control functions of the MCU are selected by the four input lines designated FC_0-FC_3. Function code formats are given in Appendix B, "Flag Control Function Summary."

The following is a detailed description of each of the eight flag control functions.

FLAG INPUT CONTROL FUNCTIONS

The flag input control functions select which flag or flags will be set to the current value of the flag input (FI) line. Data on FI is stored in the F-latch when the clock is low. The content of the F-latch is loaded into the C and/or Z flag on the rising edge of the clock.

Mnemonic	Function Description
SCZ	Set C-flag and Z-flag to FI. The C-flag and the Z-flag are both set to the value of FI.
STZ	Set Z-flag to FI. The Z-flag is set to the value of FI. The C-flag is unaffected.
STC	Set C-flag to FI. The C-flag is set to the value of FI. The Z flag is unaffected.
HCZ	Hold C-flag and Z-flag. The values in the C-flag and Z-flag are unaffected.

FLAG OUTPUT CONTROL FUNCTIONS

The flag output control functions select the value to which the flag output (FO) line will be forced.

Mnemonic	Function Description
FF0	Force FO to 0. FO is forced to the value of logical 0.
FFC	Force FO to C. FO is forced to the value of the C-flag.
FFZ	Force FO to Z. FO is forced to the value of the Z-flag.
FF1	Force FO to 1. FO is forced to the value of logical 1.

LOAD AND INTERRUPT STROBE FUNCTIONS

The load function of the MCU is controlled by the input line designated LD. If the LD line is active HIGH at the rising edge of the clock, the data on the primary and secondary instruction busses, PX_4-PX_7 and SX_0-SX_3, is loaded into the microprogram address register. PX_4-PX_7 are loaded into MA_0-MA_3 and SX_0-SX_3 are loaded into MA_4-MA_7. The high-order bit of the microprogram address register MA_8 is set to a logical 0. The bits from the primary instruction bus select 1 of 16 possible column addresses. Likewise, the bits from the secondary instruction bus select 1 of the first 16 row addresses.

The interrupt strobe enable of the MCU is available on the output line designated ISE. The line is placed in the active high state whenever a JZR to col_{15} is selected as the address control function. Customarily, the start of a macroinstruction fetch sequence is situated at row_0 and col_{15} so that the INTEL 3214 Priority Interrupt Control Unit may be enabled at the beginning of the fetch/execute cycle. The priority interrupt control unit may respond to the interrupt by pulling the enable row address (ERA) input line down to override the selected next row address from the MCU. Then by gating an alternative next row address on to the row address lines of the microprogram memory, the microprogram may be forced to enter an interrupt handling routine. The alternative row address placed on the microprogram memory address lines does not alter the contents of the microprogram address register. Therefore, subsequent jump functions will utilize the row address in the register, and not the alternative row address, to determine the next microprogram address.

Note, the load function always overrides the address control function on AC_0-AC_6. It does not, however, override the latch enable or load sub-functions of the JCE or JPX instruction, respectively. In addition, it does not inhibit the interrupt strobe enable or any of the flag control functions.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias	0°C to 70°C
Storage Temperature	–65°C to +160°C
All Output and Supply Voltages	–0.5V to +7V
All Input Voltages	–1.0V to +5.5V
Output Currents	100 mA

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

T_A = 0°C to 70°C

SYMBOL	PARAMETER	MIN	TYP(1)	MAX	UNIT	CONDITIONS
V_C	Input Clamp Voltage (All Input Pins)		–0.8	–1.0	V	V_{CC} = 4.75V, I_C = –5 mA
I_F	Input Load Current:					
	CLK Input		–0.075	–0.75	mA	V_{CC} = 5.25V, V_F = 0.45V
	EN Input		–0.05	–0.50	mA	
	All Other Inputs		–0.025	–0.25	mA	
I_R	Input Leakage Current:					
	CLK			120	μA	V_{CC} = 5.25V, V_R = 5.25V
	EN Input			80	μA	
	All Other Inputs			40	μA	
V_{IL}	Input Low Voltage			0.8	V	V_{CC} = 5.0V
V_{IH}	Input High Voltage	2.0			V	
I_{CC}	Power Supply Current		170	240	mA	V_{CC} = 5.25V(2)
V_{OL}	Output Low Voltage (All Output Pins)		0.35	0.45	V	V_{CC} = 4.75V, I_{OL} = 10 mA
V_{OH}	Output High Voltage (MA_0–MA_8, ISE, FO)	2.4	3.0		V	V_{CC} = 4.75V, I_{OH} = –1 mA
I_{OS}	Output Short Circuit Current (MA_0–MA_8, ISE, FO)	–15	–28	–60	mA	V_{CC} = 5.0V
$I_{O(off)}$	Off-State Output Current:					
	MA_0–MA_8, FO			–100	μA	V_{CC} = 5.25V, V_O = 0.45V
	MA_0–MA_8, FO, PR_0–PR_2			100	μA	V_{CC} = 5.25V, V_O = 5.25V

NOTES:
(1) Typical values are for T_A = 25°C and nominal supply voltage.
(2) EN input grounded, all other inputs and outputs open.

A.C. CHARACTERISTICS AND WAVEFORMS $T_A = 0°C$ to $70°C$, $V_{CC} = 5.0V \pm 5\%$

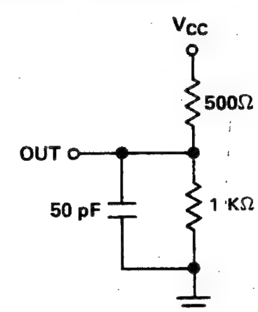
SYMBOL	PARAMETER	MIN	TYP[1]	MAX	UNIT
t_{CY}	Cycle Time	85	60		ns
t_{WP}	Clock Pulse Width	30	20		ns
	Control and Data Input Set-Up Times:				
t_{SF}	LD, AC_0–AC_6	10	0		ns
t_{SK}	FC_0, FC_1	0			ns
t_{SX}	SX_0–SX_3, PX_4–PX_7	35	25		ns
t_{SI}	FI	15	5		ns
	Control and Data Input Hold Times:				
t_{HF}	LD, AC_0–AC_6	5	0		ns
t_{HK}	FC_0, FC_1	0			ns
t_{HX}	SX_0–SX_3, PX_4–PX_7	20	5		ns
t_{HI}	FI	20	8		ns
t_{CO}	Propagation Delay from Clock Input (CLK) to Outputs (MA_0–MA_8, FO)		30	44	ns
t_{KO}	Propagation Delay from Control Inputs FC_2 and FC_3 to Flag Out (FO)		16	30	ns
t_{FO}	Propagation Delay from Control Inputs AC_0–AC_6 to Latch Outputs (PR_0–PR_2)		26	40	ns
t_{EO}	Propagation Delay from Enable Inputs EN and ERA to Outputs (MA_0–MA_8, FO, PR_0–PR_2)		21	32	ns
t_{FI}	Propagation Delay from Control Inputs AC_0–AC_6 to Interrupt Strobe Enable Output (ISE)		24	40	ns

NOTE:
(1) Typical values are for $T_A = 25°C$ and nominal supply voltage.

TEST CONDITIONS:

Input pulse amplitude of 2.5 volts.
Input rise and fall times of 5 ns between 1 volt and 2 volts.
Output load of 10 mA and 50 pF.
Speed measurements are taken at the 1.5 volt level.

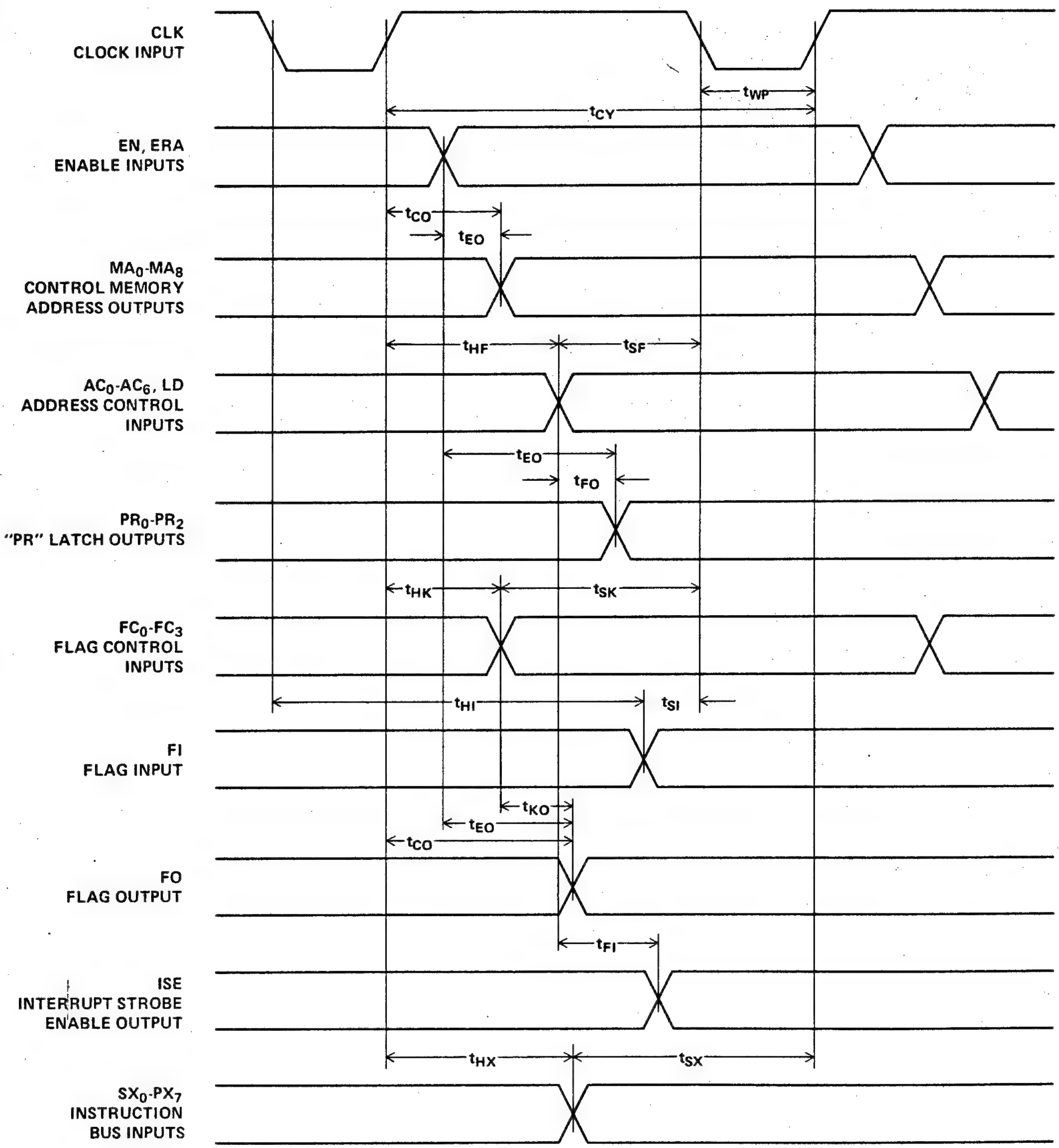
TEST LOAD CIRCUIT:



CAPACITANCE[2] $T_A = 25°C$

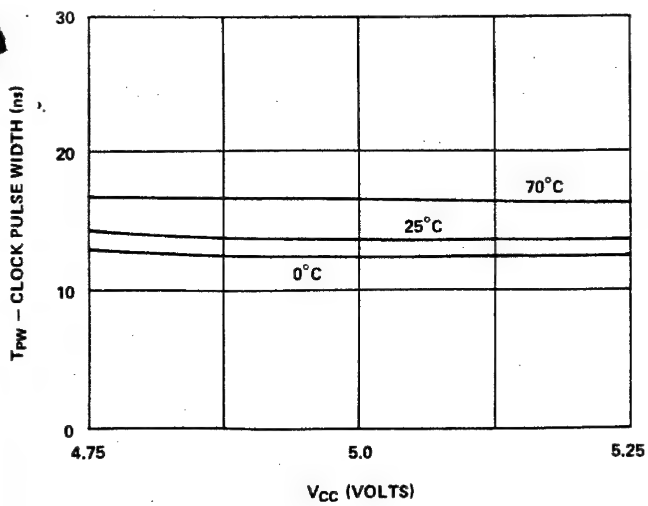
SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT
C_{IN}	Input Capacitance:				
	CLK, EN		11	16	pF
	All Other Inputs		5	10	pF
C_{OUT}	Output Capacitance		6	12	pF

NOTE:
(2) This parameter is periodically sampled and is not 100% tested. Condition of measurement is f = 1 MHz, $V_{BIAS} = 2.5V$, $V_{CC} = 5V$ and $T_A = 25°C$.

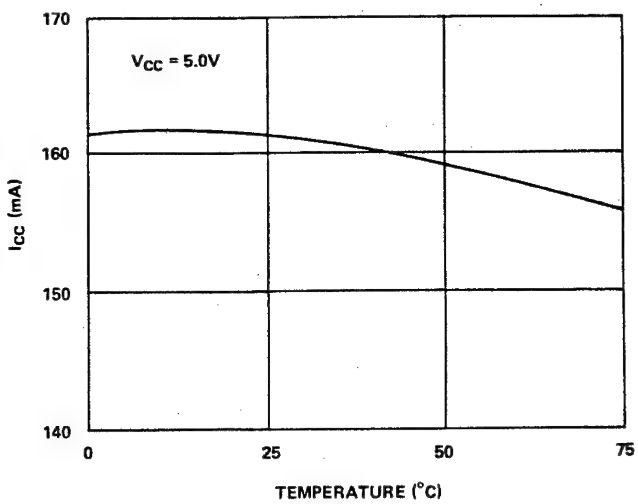

CLK
CLOCK INPUT
tWP
tCY
EN, ERA
ENABLE INPUTS
tCO
tEO
MA0-MA8
CONTROL MEMORY
ADDRESS OUTPUTS
tHF
tSF
AC0-AC6, LD
ADDRESS CONTROL
INPUTS
tEO
tFO
PR0-PR2
"PR" LATCH OUTPUTS
tHK
tSK
FC0-FC3
FLAG CONTROL
INPUTS
tHI
tSI
FI
FLAG INPUT
tKO
tEO
tCO
FO
FLAG OUTPUT
tFI
ISE
INTERRUPT STROBE
ENABLE OUTPUT
tHX
tSX
SX0-PX7
INSTRUCTION
BUS INPUTS

TYPICAL AC AND DC CHARACTERISTICS

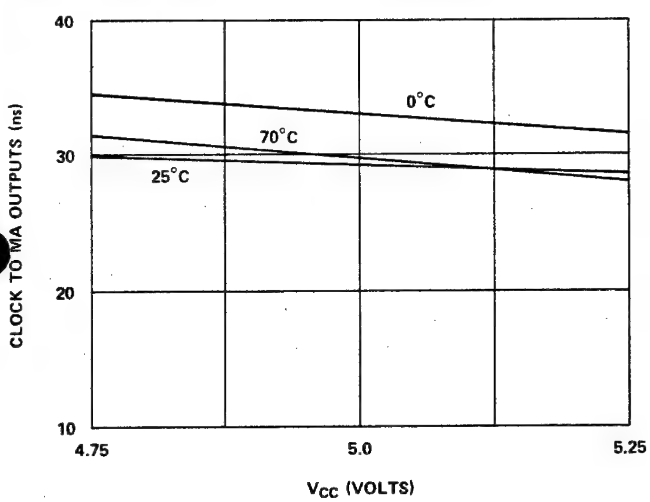
Clock Pulse Width vs V_{CC} and Temperature



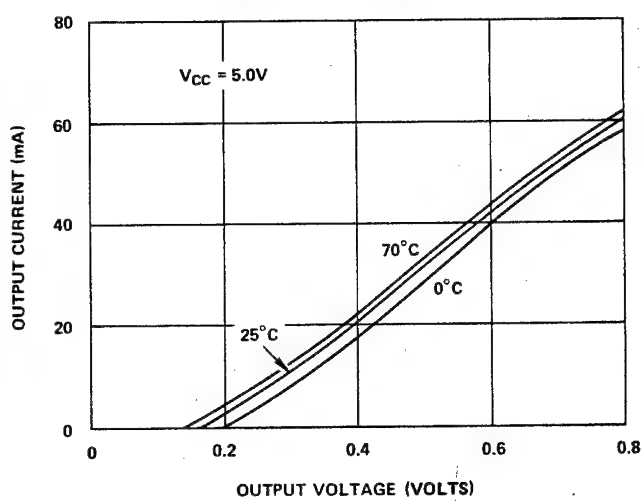
I_{CC} vs Temperature



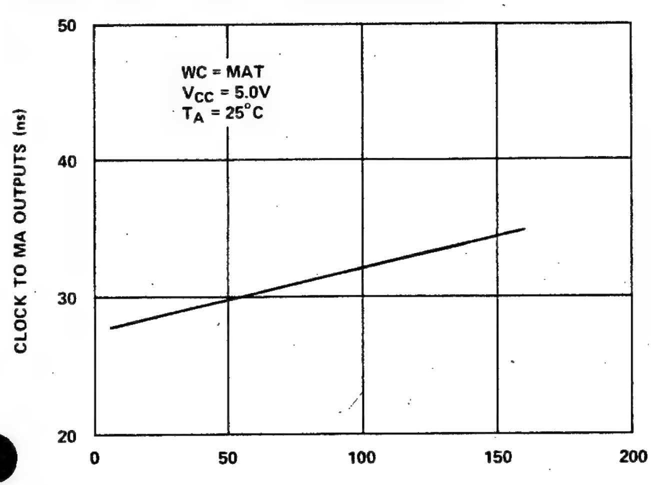
Clock to MA Outputs vs V_{CC} and Temperature



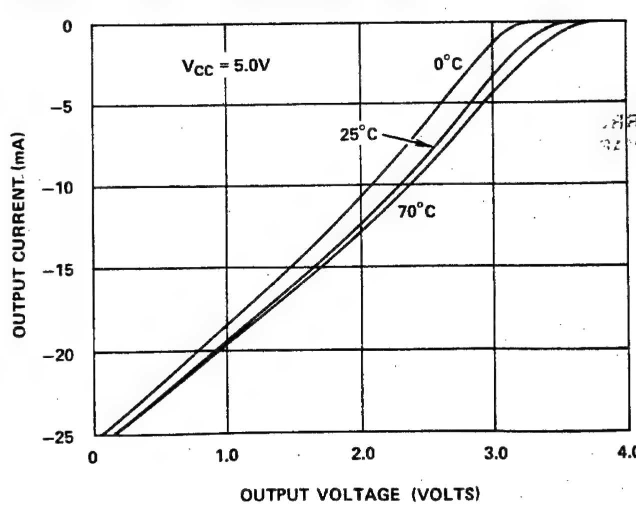
Output Current vs Output Low Voltage



Clock to MA Outputs vs Load Capacitance



Output Current vs Output High Voltage



APPENDIX A ADDRESS CONTROL FUNCTION SUMMARY

MNEMONIC	DESCRIPTION	FUNCTION AC_6	5	4	3	2	1	0	NEXT ROW MA_8	7	6	5	4	NEXT COL MA_3	2	1	0
JCC	Jump in current column	0	0	d_4	d_3	d_2	d_1	d_0	d_4	d_3	d_2	d_1	d_0	m_3	m_2	m_1	m_0
JZR	Jump to zero row	0	1	0	d_3	d_2	d_1	d_0	0	0	0	0	0	d_3	d_2	d_1	d_0
JCR	Jump in current row	0	1	1	d_3	d_2	d_1	d_0	m_8	m_7	m_6	m_5	m_4	d_3	d_2	d_1	d_0
JCE	Jump in column/enable	1	1	1	0	d_2	d_1	d_0	m_8	m_7	d_2	d_1	d_0	m_3	m_2	m_1	m_0
JFL	Jump/test F-latch	1	0	0	d_3	d_2	d_1	d_0	m_8	d_3	d_2	d_1	d_0	m_3	0	1	f
JCF	Jump/test C-flag	1	0	1	0	d_2	d_1	d_0	m_8	m_7	d_2	d_1	d_0	m_3	0	1	c
JZF	Jump/test Z-flag	1	0	1	1	d_2	d_1	d_0	m_8	m_7	d_2	d_1	d_0	m_3	0	1	z
JPR	Jump/test PR-latches	1	1	0	0	d_2	d_1	d_0	m_8	m_7	d_2	d_1	d_0	p_3	p_2	p_1	p_0
JLL	Jump/test left PR bits	1	1	0	1	d_2	d_1	d_0	m_8	m_7	d_2	d_1	d_0	0	1	p_3	p_2
JRL	Jump/test right PR bits	1	1	1	1	1	d_1	d_0	m_8	m_7	1	d_1	d_0	1	1	p_1	p_0
JPX	Jump/test PX-bus	1	1	1	1	0	d_1	d_0	m_8	m_7	m_6	d_1	d_0	x_7	x_6	x_5	x_4

SYMBOL	MEANING
d_n	Data on address control line n
m_n	Data in microprogram address register bit n
p_n	Data in PR-latch bit n
x_n	Data on PX-bus line n (active LOW)
f, c, z	Contents of F-latch, C-flag, or Z-flag, respectively

APPENDIX B FLAG CONTROL FUNCTION SUMMARY

TYPE	MNEMONIC	DESCRIPTION	FC_1	0
Flag Input	SCZ	Set C-flag and Z-flag to f	0	0
	STZ	Set Z-flag to f	0	1
	STC	Set C-flag to f	1	0
	HCZ	Hold C-flag and Z-flag	1	1

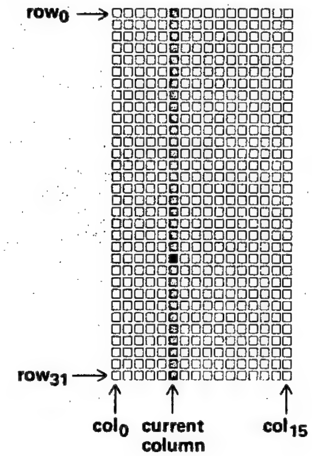
TYPE	MNEMONIC	DESCRIPTION	FC_3	2
Flag Output	FF0	Force FO to 0	0	0
	FFC	Force FO to C-flag	0	1
	FFZ	Force FO to Z-flag	1	0
	FF1	Force FO to 1	1	1

LOAD FUNCTION LD	NEXT ROW MA_8	7	6	5	4	NEXT COL MA_3	2	1	0
0	see Appendix A					see Appendix A			
1	0	x_3	x_2	x_1	x_0	x_7	x_6	x_5	x_4

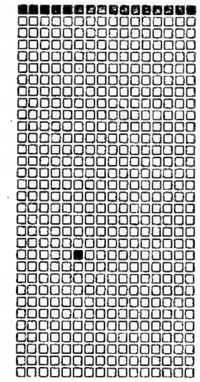
SYMBOL	MEANING
f	Contents of the F-latch
x_n	Data on PX- or SX-bus line n (active LOW)

The following ten diagrams illustrate the jump set for each of the eleven jump and jump/test functions of the MCU. Location 341, indicated by the black square, represents one current row (row_{21}) and current column (col_5) address. The blue boxes indicate the microprogram locations that may be selected by the particular function as the next address.

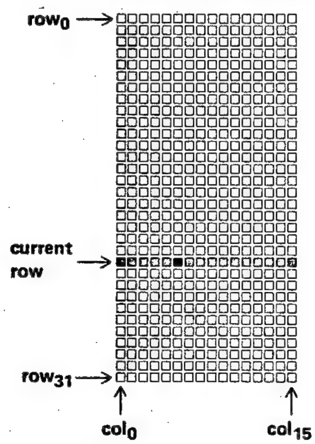
JCC
Jump in Current Column



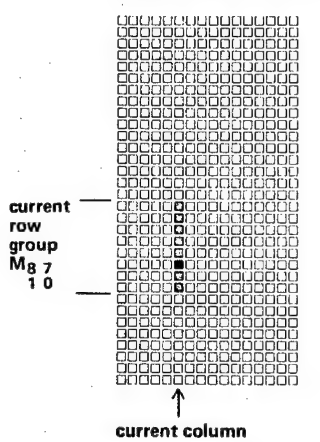
JZR
Jump to Zero Row



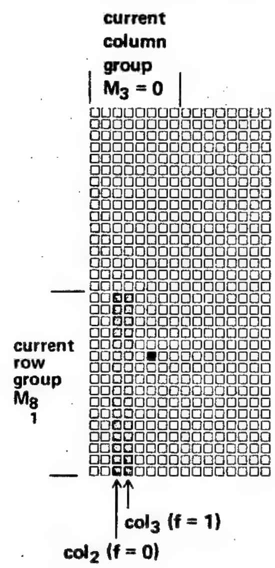
JCR
Jump in Current Row



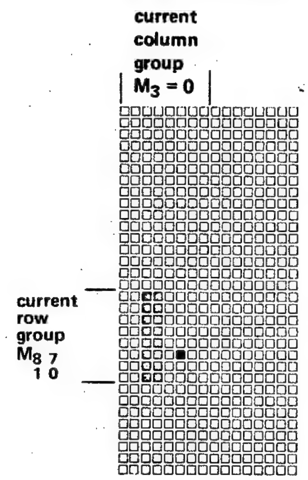
JCE
Jump Column/Enable



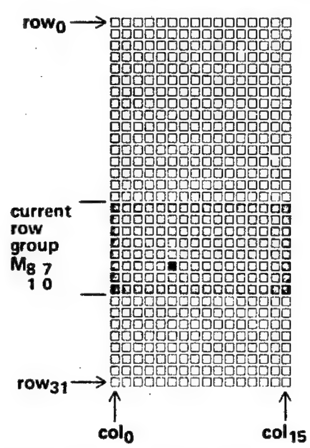
JFL
Jump/Test F-Latch



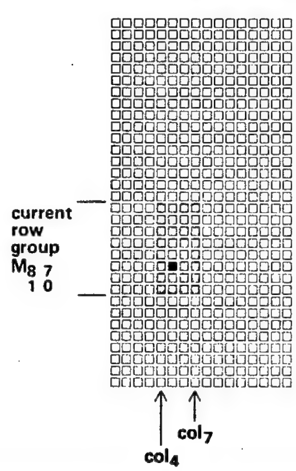
JCF, JZF
Jump/Test C-Flag
Jump/Test Z-Flag



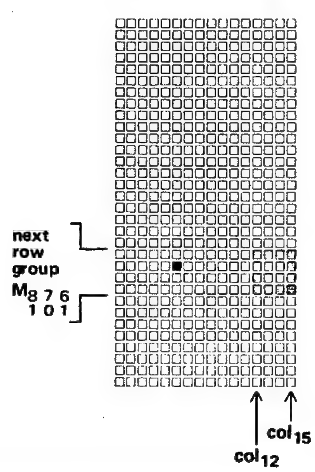
JPR
Jump/Test PR-Latch



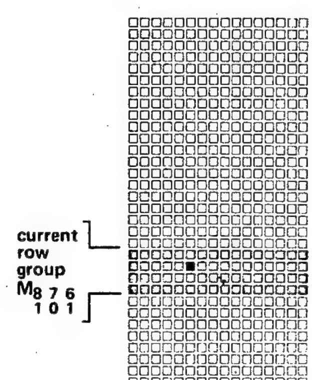
JLL
Jump/Test Left Latch

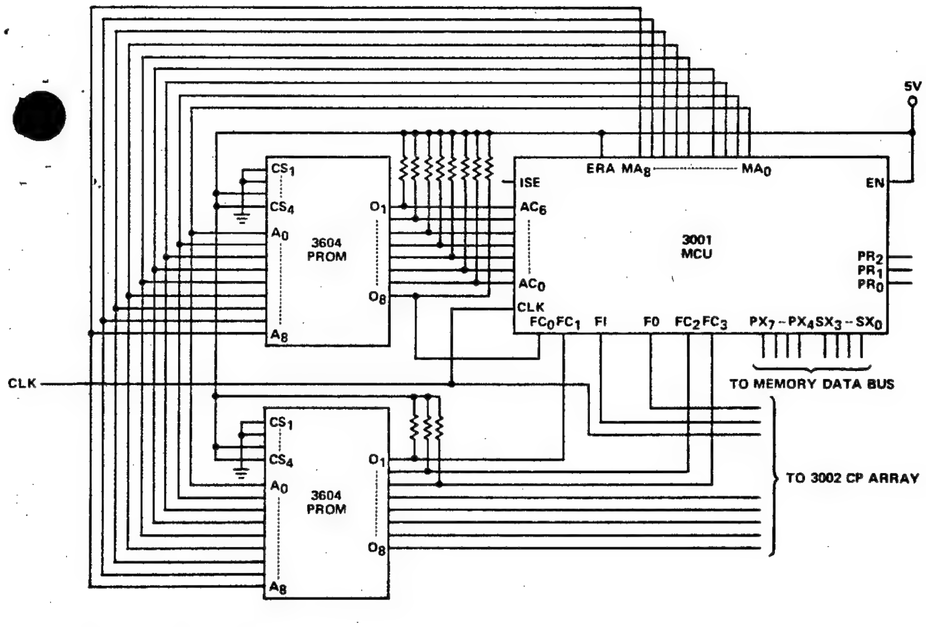


JRL
Jump/Test Right Latch

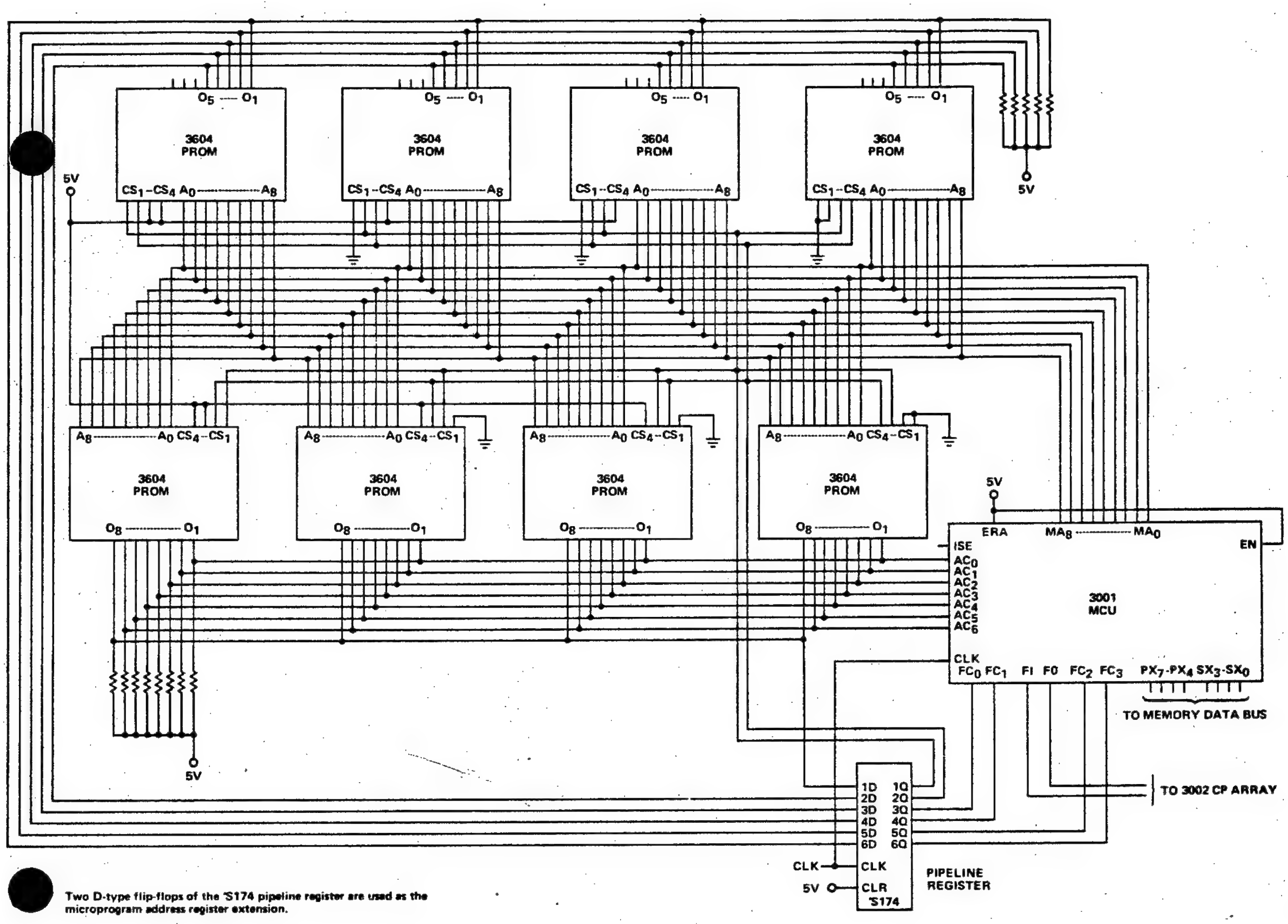


JPX
Jump/Test PX-Bus





Non-Pipelined Configuration with
512 Microinstruction Addressability



Two D-type flip-flops of the 'S174 pipeline register are used as the microprogram address register extension.

Pipelined Configuration with
2048 Microinstruction Addressability

ORDERING INFORMATION:

Part Number	Description
C3001	Microprogram Control Unit

intel®

Intel Corporation
3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 246-7501
TWX: 910-338-0026
Telex: 34-6372

WESTERN
1651 East 4th Street
Suite 228
Santa Ana, California 92701
Tel: (714) 835-9642
TWX: 910-595-1114

MID-AMERICA
6350 L.B.J. Freeway
Suite 178
Dallas, Texas 75240
Tel: (214) 661-8829
TWX: 910-860-5487

GREAT LAKES REGION
8312 North Main Street
Dayton, Ohio 45415
Tel: (513) 890-5350
TELEX: 288-004

EASTERN
2 Militia Drive
Suite 4
Lexington, Massachusetts 02173
Tel: (617) 861-1136
TWX: 710-321-0187

MID-ATLANTIC
520 Pennsylvania Avenue
Suite 102
Fort Washington, Pennsylvania 19034
Tel: (215) 542-9444
TWX: 510-661-3055

EUROPE
Belgium
Intel Office
216 Avenue Louise
Brussels B1050
Tel: 649-20-03
TELEX: 24814

ORIENT
Japan
Intel Japan Corporation
Kasahara Bldg.
1-6-10, Uchikanda
Chiyoda-ku
Tokyo 101
Tel: (03) 295-5441
TELEX: 781-28426

Printed in U.S.A. MCS-268-0275/27.5K

SCHOTTKY BIPOLAR LSI MICROCOMPUTER SET

3002 CENTRAL PROCESSING ELEMENT

The INTEL® Bipolar Microcomputer Set is a family of Schottky bipolar LSI circuits which simplify the construction of microprogrammed central processors and device controllers. These processors and controllers are truly microprogrammed in the sense that their control logic is organized around a separate read-only memory called the microprogram memory. Control signals for the various processing elements are generated by the microinstructions contained in the microprogram memory. In the implementation of a typical central processor, as shown below, the microprogram interprets a higher level of instructions called macroinstructions, similar to those found in a small computer. For device controllers, the microprograms directly implement the required control functions.

The INTEL® 3002 Central Processing Element contains all of the circuits that represent a 2-bit wide slice through the data processing section of a digital computer. To construct a complete central processor for a given word width N, it is simply necessary to connect an array of N/2 CPE's together. When wired together in such an array, a set of CPE's provide the following capabilities:

- 2's complement arithmetic
- Logical AND, OR, NOT and exclusive-OR
- Incrementing and decrementing
- Shifting left or right
- Bit testing and zero detection
- Carry look-ahead generation
- Multiple data and address busses

High Performance — 100 ns Cycle Time

TTL and DTL Compatible

N-Bit Word Expandable Multi-Bus Organization
- **3 Input Data Busses**
- **2 Three-State Fully Buffered Output Data Busses**

11 General Purpose Registers

Full Function Accumulator

Independent Memory Address Register

Cascade Outputs for Full Carry Look-Ahead

Versatile Functional Capability
- **8 Function Groups**
- **Over 40 Useful Functions**
- **Zero Detect and Bit Test**

Single Clock

28 Pin DIP

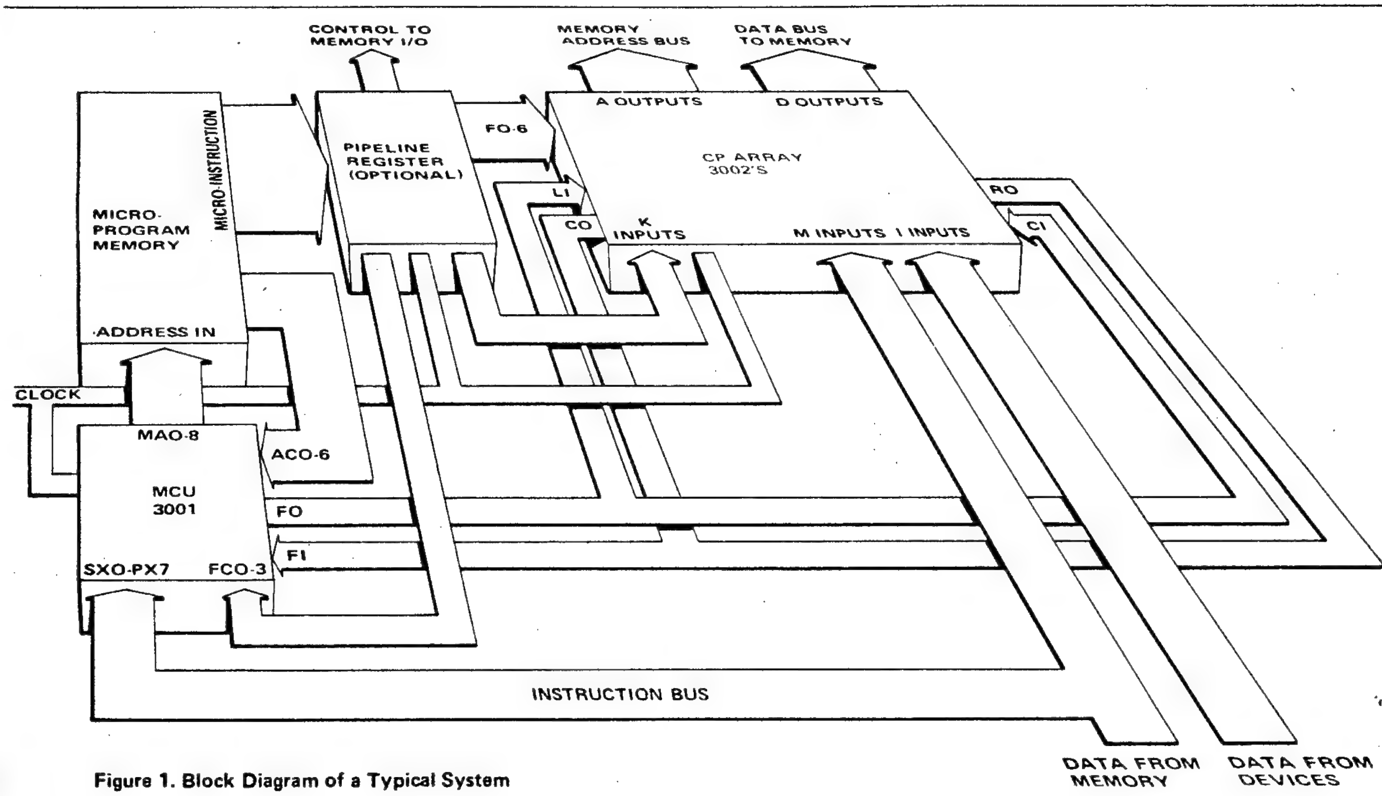


Figure 1. Block Diagram of a Typical System

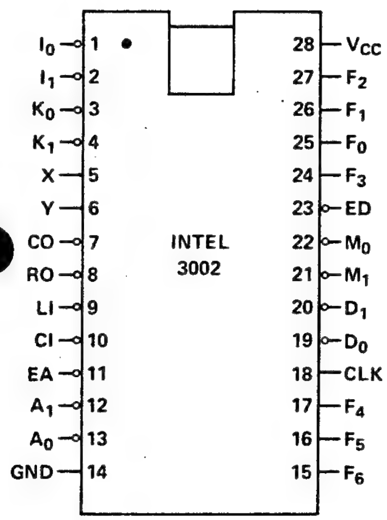
Other members of the INTEL Bipolar Microcomputer Set:

3001 Microprogram Control Unit	3214 Priority Interrupt Control Unit	3304A Schottky Bipolar ROM (512 x 8)
3003 Look-Ahead Carry Generator	3226 Inverting Bi-Directional Bus Driver	3601 Schottky Bipolar PROM (256 x 4)
3212 Multi-Mode Latch Buffer	3301 Schottky Bipolar ROM (256 x 4)	3604 Schottky Bipolar PROM (512 x 8)

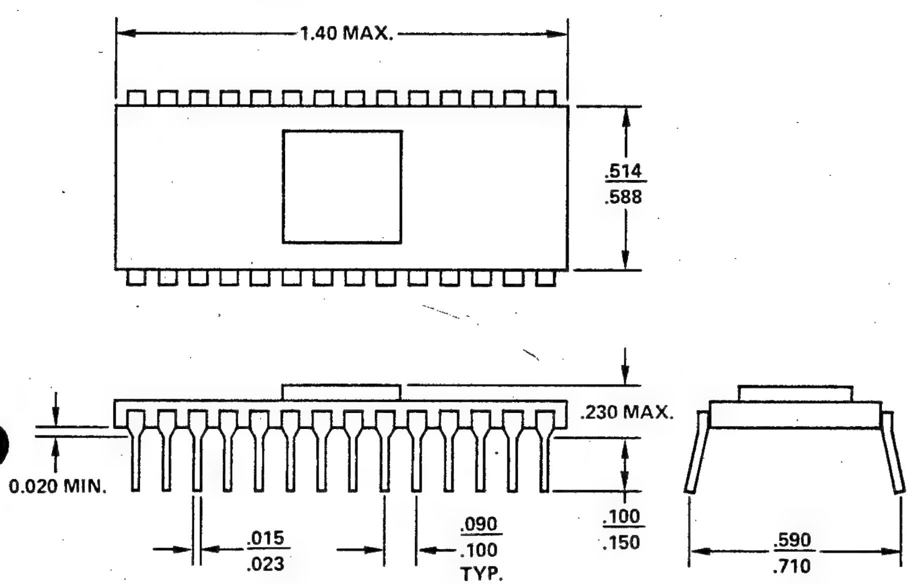
CONTENTS

APPENDICES

PACKAGE CONFIGURATION



PACKAGE OUTLINE



PIN DESCRIPTION

PIN	SYMBOL	NAME AND FUNCTION	TYPE[1]
1, 2	I_0-I_1	External Bus Inputs The external bus inputs provide a separate input port for external input devices.	Active LOW
3, 4	K_0-K_1	Mask Bus Inputs The mask bus inputs provide a separate input port for the microprogram memory, to allow mask or constant entry.	Active LOW
5, 6	X, Y	Standard Carry Look-Ahead Cascade Outputs The cascade outputs allow high speed arithmetic operations to be performed when they are used in conjunction with the INTEL 3003 Look-Ahead Carry Generator.	
7	CO	Ripply Carry Output The ripple carry output is only disabled during shift right operations.	Active LOW Three-state
8	RO	Shift Right Output The shift right output is only enabled during shift right operations.	Active LOW Three-state
9	LI	Shift Right Input	Active LOW
10	CI	Carry Input	Active LOW
11	EA	Memory Address Enable Input When in the LOW state, the memory address enable input enables the memory address outputs (A_0-A_1).	Active LOW
12-13	A_0-A_1	Memory Address Bus Outputs The memory address bus outputs are the buffered outputs of the memory address register (MAR).	Active LOW Three-state
14	GND	Ground	
15-17, 24-27,	F_0-F_6	Micro-Function Bus Inputs The micro-function bus inputs control ALU function and register selection.	
18	CLK	Clock Input	
19-20	D_0-D_1	Memory Data Bus Outputs The memory data bus outputs are the buffered outputs of the full function accumulator register (AC).	Active LOW Three-state
21-22	M_0-M_1	Memory Data Bus Inputs The memory data bus inputs provide a separate input port for memory data.	Active LOW
23	ED	Memory Data Enable Input When in the LOW state, the memory data enable input enables the memory data outputs (D_0-D_1)	Active LOW
28	V_{CC}	+5 Volt Supply	

NOTE:
1. Active HIGH, unless otherwise specified.

The CPE provides the arithmetic, logic and register functions of a 2-bit wide slice through a microprogrammed central processor. Data from external sources such as main memory, is brought into the CPE on one of the three separate input busses. Data being sent out of the CPE to external devices is carried on either of the two output busses. Within the CPE, data is stored in one of eleven scratchpad registers or in the accumulator. Data from the input busses, the registers, or the accumulator is available to the arithmetic/logic section (ALS) under the control of two internal multiplexers. Additional inputs and outputs are included for carry propagation, shifting, and micro-function selection. The complete logical organization of the CPE is shown below.

MICRO-FUNCTION BUS AND DECODER

The seven micro-function bus input lines of the CPE, designated F_0–F_6, are decoded internally to select the ALS function, generate the scratchpad address, and control the A and B multiplexers.

M-BUS AND I-BUS INPUTS

The M-bus inputs are arranged to bring data from an external main memory into the CPE. Data on the M-bus is multiplexed internally for input to the ALS.

The I-bus inputs are arranged to bring data from an external I/O system into the CPE. Data on the I-bus is also multiplexed internally, although independently of the M-bus, for input to the ALS Separation of the two busses permits a relatively lightly loaded memory bus even though a large number of I/O devices are connected to the I-bus. Alternatively, the I-bus may be wired to perform a multiple bit shift (e.g., a byte exchange) by connecting it to one of the output busses. In this case, I/O device data is gated externally onto the M-bus.

SCRATCHPAD

The scratchpad contains eleven registers designated R_0 through R_9 and T. The output of the scratchpad is multiplexed internally for input to ALS. The ALS output is returned for input into the scratchpad.

ACCUMULATOR AND D-BUS

An independent register called the accumulator (AC) is available for storing the result of an ALS operation. The output of the accumulator is multiplexed internally for input back to the ALS and is also available via a three-state output buffer on the D-bus outputs. Conventional usage of the D-bus is for data being sent to the external main memory or to external I/O devices.

A AND B MULTIPLEXERS

The A and B multiplexers select the two inputs to the ALS specified on the micro-function bus. Inputs to the A-multiplexer include the M-bus, the scratchpad, and the accumulator. The B-multiplexer selects either the I-bus, the accumulator, or the K-bus. The selected B-multiplexer input is always logically ANDed with the data on the K-bus (see below) to provide a flexible masking and bit testing capability.

ALS AND K-BUS

The ALS is capable of a variety of arithmetic and logic operations, including 2's complement addition, incrementing, and decrementing, plus logical AND, inclusive-OR, exclusive-NOR, and logical complement. The result of an ALS operation may be stored in the accumulator or one of the scratchpad registers. Separate left input and right output lines, designated LI and RO, are available for use in right shift operations. Carry input and carry output lines, designated CI and CO are provided for normal ripple carry propagation. CO and RO data are brought out via two alternately enabled tri-state buffers. In addition, standard look ahead carry outputs, designated X and Y, are available for full carry look ahead across any word length.

The ability of the K-bus to mask inputs to the ALS greatly increases the versatility of the CPE. During non-arithmetic operations in which carry propagation has no meaning, the carry circuits are used to perform a word-wise inclusive-OR of the bits, masked by the K-bus, from the register or bus selected by the function decoder. Thus, the CPE provides a flexible bit testing capability. The K-bus is also used during arithmetic operations to mask portions of the field being operated upon. An additional function of the K-bus is that of supplying constants to the CPE from the microprogram.

MEMORY ADDRESS REGISTER AND A-BUS

A separate ALS output is also available to the memory address register (MAR) and to the A-bus via a three-state output buffer. Conventional usage of the MAR and A-bus is for sending addresses to an external main memory. The MAR and A-bus may also be used to select an external device when executing I/O operations.

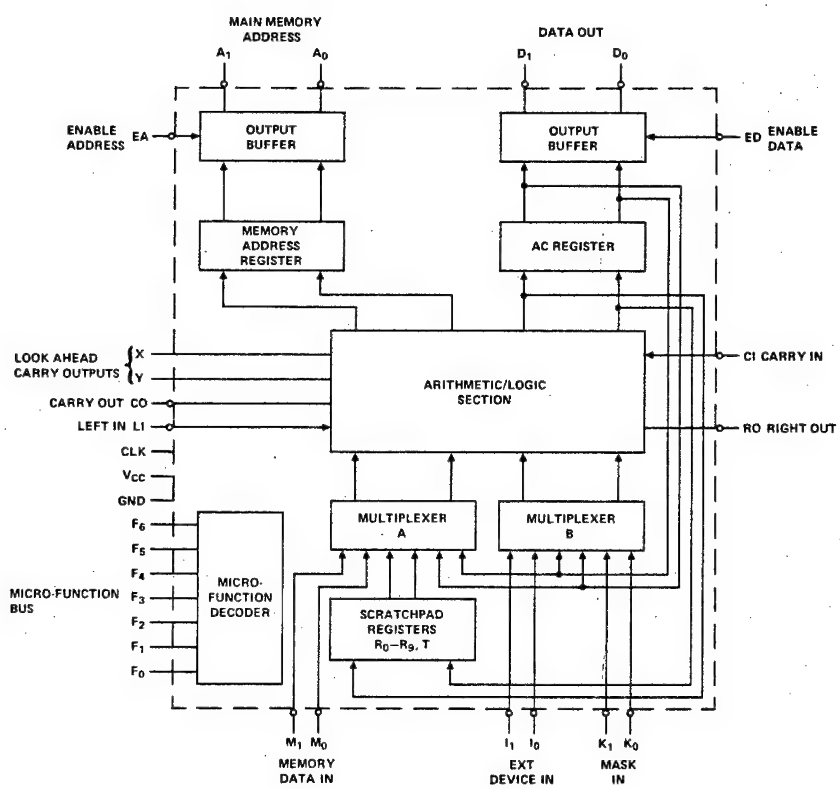


Figure 2. 3002 Block Diagram

During each micro-cycle, a micro-function is applied to F-bus inputs of the CPE. The micro-function is decoded, the operands are selected by the multiplexers, and the specified operation is performed by ALS. If a negative going clock edge is applied, the result of the ALS operation is either deposited in the accumulator or written into the selected scratchpad register. In addition, certain operations permit related address data to be deposited in the MAR. A new micro-function should only be applied following the rising edge of the clock.

By externally gating the clock input to CPE, referred to as conditional clocking, the clock pulse may be selectively omitted during a micro-cycle. Since the carry, shift, and look-ahead circuits are not clocked, their outputs may be used to perform a variety of non-destructive tests on data in the accumulator or in the scratchpad. No register contents are modified by the operation due to the absence of the clock pulse.

The micro-function to be performed is determined from the function group (F-Group) and register group (R-Group) selected by the data on the F-bus. The F-Group is specified by the upper three bits of data, F_4-F_6. The R-Group is specified by the lower four bits of data, F_0-F_3. R-Group I contains R_0 through R_9, T, and AC and is denoted by the symbol R_n. R-Group II and R-Group III contain only T and AC. F-Group and R-Group formats are summarized in Appendix A.

The following is a detailed explanation of each of the CPE micro-functions. A general functional description of each operation is given followed by two additional descriptions which explain the result of the micro-function with both K-bus inputs at logical 0 or both at logical 1. In most cases, the effect of placing the K-bus in the all-one or the all-zero state is to either select or deselect the accumulator in the operation, respectively. A micro-function mnemonic is included with each description for reference purposes and to assist in the design of micro-assembly languages. The micro-functions are summarized in Appendix B. The effective micro-functions for the all-zero and the all-one K-bus states are summarized in Appendix C and D, respectively.

F-GROUP 0 — R-GROUP I

Logically AND the contents of AC with the data on the K-bus. Add the result to the contents of R_n and the value of the carry input (CI). Deposit the sum in AC and R_n.

ILR — K-BUS = 00

Conditionally increment R_n and load the result in AC. Used to load AC from R_n or to increment R_n and load a copy of the result in AC.

ALR — K-BUS= 11

Add AC and CI to R_n and load the result in AC. Used to add AC to a register. If R_n is AC, then AC is shifted left one bit position.

F-GROUP 0 — R-GROUP II

Logically AND the contents of AC with the data on the K-bus. Add the result to CI and the data on the M-bus. Deposit the sum in AC or T, as specified.

ACM — K-BUS = 00

Add CI to the data on the M-bus. Load the result in AC or T, as specified. Used to load memory data in the specified register, or to load incremented memory data in the specified register.

AMA — K-BUS = 11

Add the data on the M-bus to AC and CI, and load the result in AC or T, as specified. Used to add memory data or incremented memory data to AC and store the sum in the specified register.

F-GROUP 0 — R-GROUP III

(General description omitted, see Appendix B.)

SRA — K-BUS = 00

Shift the contents of AC or T, as specified, right one bit position. Place the previous low order bit value on RO and fill the high order bit from the data on LI. Used to shift or rotate AC or T right one bit.

(K-bus = 11 description omitted, see Appendix B.)

F-GROUP 1 — R-GROUP I

Logically OR the contents of R_n with the data on the K-bus. Deposit the result in MAR. Add the data on the K-bus to contents of R_n and CI. Deposit the result in R_n.

LMI — K-BUS = 00

Load MAR from R_n. Conditionally increment R_n. Used to maintain a macro-instruction program counter.

DSM — K-BUS = 11

Set MAR to all one's. Conditionally decrement R_n by one. Used to force MAR to its highest address and to decrement R_n.

F-GROUP 1 — R-GROUP II

Logically OR the data on the M-bus with the data on the K-bus. Deposit the result in MAR. Add the data on the K-bus to the data on the M-bus and CI. Deposit the sum in AC or T, as specified.

LMM — K-BUS = 00

Load MAR from the M-bus. Add CI to the data on the M-bus. Deposit the result in AC or T. Used to load the address register with memory data for macro-instructions using indirect addressing.

LDM — K-BUS = 11

Set MAR to all ones. Subtract one from the data on the M-bus. Add CI to the difference and deposit the result in AC or T, as specified. Used to load decremented memory data in AC or T.

F-GROUP 1 — R-GROUP III

Logically OR the data on the K-bus with the complement of the contents of AC or T, as specified. Add the result to the logical AND of the contents of specified register with the data on the K-bus. Add the sum to CI. Deposit the result in the specified register.

CIA — K-BUS = 00

Add CI to the complement of the contents of AC or T, as specified. Deposit the result in the specified register. Used to form the 1's or 2's complement of AC or T.

DCA — K-BUS = 11

Subtract one from the contents of AC or T, as specified. Add CI to the difference and deposit the sum in the specified register. Used to decrement AC or T.

F-GROUP 2 R-GROUP I

Logically AND the data on the K-bus with the contents of AC. Subtract one from the result and add the difference to CI. Deposit the sum in R_n.

CSR K-BUS = 00

Subtract one from CI and deposit the difference in R_n. Used to conditionally clear or set R_n to all 0's or 1's, respectively.

SDR K-BUS = 11

Subtract one from AC and add the difference to CI. Deposit the sum in R_n. Used to store AC in R_n or to store the decremented value of AC in R_n.

F-GROUP 2 R-GROUP II

Logically AND the data on the K-bus with the contents of AC. Subtract one from the result and add the difference to CI. Deposit the sum in AC or T, as specified.

CSA K-BUS = 00

Subtract one from CI and deposit the difference in AC or T, as specified. Used to conditionally clear or set AC or T.

SDA K-BUS = 11

Subtract one from AC and add the difference to CI. Deposit the sum in AC or T, as specified. Used to store AC in T, or decrement AC, or store the decremented value of AC in T.

F-GROUP 2 R-GROUP III

Logically AND the data of the K-bus with the data on the I-bus. Subtract one from the result and add the difference to CI. Deposit the sum in AC or T, as specified.

(K-bus = 00 description omitted, see CSA above.)

LDI K-BUS = 11

Subtract one from the data on the I-bus and add the difference to CI. Deposit the sum in AC or T, as specified. Used to load input bus data or decremented input bus data in the specified register.

F-GROUP 3 R-GROUP I

Logically AND the contents of AC with the data on the K-bus. Add the contents of R_n and CI to the result. Deposit the sum in R_n.

INR K-BUS = 00

Add CI to the contents of R_n and deposit the sum in R_n. Used to increment R_n.

ADR K-BUS = 11

Add the contents of AC to R_n. Add the result to CI and deposit the sum in R_n. Used to add the accumulator to a register or to add the incremented value of the accumulator to a register.

F-GROUP 3 R-GROUP II

(All descriptions omitted, identical to F-Group 0/R-Group II described above.)

F-GROUP 3 R-GROUP III

Logically AND the data on the K-bus with the data on the I-bus. Add CI and the contents of AC or T, as specified, to the result. Deposit the sum in the specified register.

INA K-BUS = 00

Conditionally increment the contents of AC or T, as specified. Used to increment AC or T.

AIA K-BUS = 11

Add the data on the I-bus to the contents of AC or T, as specified. Add CI to the result and deposit the sum in the specified register. Used to add input data or incremented input data to the specified register.

F-GROUP 4 R-GROUP I

Logically AND the data on the K-bus with the contents of AC. Logically AND the result with the contents of R_n. Deposit the final result in R_n. Logically OR the value of CI with the word-wise OR of the bits of the final result. Place the value of the carry OR on the carry output (CO) line.

CLR K-BUS = 00

Clear R_n to all 0's. Force CO to CI. Used to clear a register and force CO to CI.

ANR K-BUS = 11

Logically AND AC with R_n. Deposit the result in R_n. Force CO to one if the result is non-zero. Used to AND the accumulator with a register and test for a zero result.

F-GROUP 4 R-GROUP II

Logically AND the data on the K-bus with the contents of AC. Logically AND the result with the data on the M-bus. Deposit the final result in AC or T, as specified. Logically OR the value of CI with the word-wise OR of the bits of the final result. Place the value of the carry OR on CO.

CLA K-BUS = 00

Clear AC or T, as specified, to all 0's. Force CO to CI. Used to clear the specified register and force CO to CI.

ANM K-BUS = 11

Logically AND the data on the M-bus with the contents of AC. Deposit the result in AC or T, as specified. Force CO to one if the result is non-zero. Used to AND M-bus data to the accumulator and test for a zero result.

F-GROUP 4 R-GROUP III

Logically AND the data on I-bus with the data on the K-bus. Logically AND the result with the contents of AC or T, as specified. Deposit the final result in the specified register. Logically OR CI with the word-wise OR of the bits of the final result. Place the value of the carry OR on CO.

(K-bus = 00 description omitted, see CLA above.)

ANI K-BUS = 11

Logically AND the data on the I-bus with the contents of AC or T, as specified. Deposit the result in the specified register. Force CO to one if the result is non-zero. Used to AND the I-bus to the accumulator and test for a zero result.

F-GROUP 5 R-GROUP I

Logically AND the data on the K-bus with the contents of R_n. Deposit the result in R_n. Logically OR CI with the word-wise OR of the result. Place the value of the carry OR on CO.

(K-bus = 00 description omitted, see CLR above.)

TZR K-BUS = 11

Force CO to one if R_n is non-zero. Used to test a register for zero. Also used to AND K-bus data with a register (see general description) for masking and, optionally, testing for a zero result.

F-GROUP 5 R-GROUP II

Logically AND the data on the K-bus with the data on the M-bus. Deposit the result in AC or T, as specified. Logically OR CI with the word-wise OR of the result. Place the value of the carry OR on CO.

(K-bus = 00 description omitted, see CLA above.)

LTM K-BUS = 11

Load AC or T, as specified, with data from the M-bus. Force CO to one if the result is non-zero. Used to load the specified register from memory and test for a zero result. Also used to AND K-bus data with M-bus data (see general description) for masking and, optionally, testing for a zero result.

F-GROUP 5 R-GROUP III

Logically AND the data on K-bus with contents of AC or T, as specified. Deposit the result in the specified register. Logically OR CI with the word-wise OR of the result. Place the value of the carry OR on CO.

(K-bus = 00 description omitted, see CLA above.)

TZA K-BUS = 11

Force CO to one if AC or T, as specified, is non-zero. Used to test the specified register for zero. Also used to AND K-bus data to the specified register (see general description) for masking and, optionally, testing for a zero result.

F-GROUP 6 R-GROUP I

Logically OR CI with the word-wise OR of the logical AND of AC and the data on the K-bus. Place the result of the carry OR on CO. Logically OR the contents of R_n with the logical AND of AC and the data on the K-bus. Deposit the result in R_n.

NOP K-BUS = 00

Force CO to CI. Used as a null operation or to force CO to CI.

ORR K-BUS = 11

Force CO to one if AC is non-zero. Logically OR the contents of the accumulator to the contents of R_n. Deposit the result in R_n. Used to OR the accumulator to a register and, optionally, test the previous accumulator value for zero.

F-GROUP 6 R-GROUP II

Logically OR CI with the word-wise OR of the logical AND of AC and the data on the K-bus. Place the value of the carry OR on CO. Logically OR the data on the M-bus, with the logical AND of AC and the data on the K-bus. Deposit the final result in AC or T, as specified.

LMF K-BUS = 00

Load AC or T, as specified, from the M-bus. Force CO to CI. Used to load the specified register with memory data and force CO to CI.

ORM K-BUS = 11

Force CO to one if AC is non-zero. Logically OR the data on the M-bus with the contents of AC. Deposit the result in AC or T, as specified. Used to OR memory data with the accumulator and, optionally, test the previous value of the accumulator for zero.

F-GROUP 6 R-GROUP III

Logically OR CI with the word-wise OR of the logical AND of the data on the I-bus and the data on the K-bus. Place the value of the carry OR on CO. Logically AND the data on the K-bus with the data on the I-bus. Logically OR the result with the contents of AC or T, as specified. Deposit the final result in the specified register.

(K-bus = 00 description omitted, see NOP above.)

ORI K-BUS = 11

Force CO to one if the data on the I-bus is non-zero. Logically OR the data on the I-bus to the contents of AC or T, as specified. Deposit the result in the specified register. Used to OR I-bus data with the specified register and, optionally, test the I-bus data for zero.

F-GROUP 7 R-GROUP I

Logically OR CI with the word-wise OR of the logical AND of the contents of R_n and AC and the data on the K-bus. Place the value of the carry OR on CO. Logically AND the data on the K-bus with the contents of AC. Exclusive-NOR the result with the contents of R_n. Deposit the final result in R_n.

CMR K-BUS = 00

Complement the contents of R_n. Force CO to CI.

XNR K-BUS = 11

Force CO to one if the logical AND of AC and R_n is non-zero. Exclusive-NOR the contents of AC with the contents of R_n. Deposit the result in R_n. Used to exclusive-NOR the accumulator with a register.

F-GROUP 7 R-GROUP II

Logically OR CI with the word-wise OR of the logical AND of the contents of AC and the data on the K-bus and M-bus. Place the value of the carry OR on CO. Logically AND the data on the K-bus with the contents of AC. Exclusive-NOR the result with the data on the M-bus. Deposit the final result in AC or T, as specified.

LCM K–BUS = 00

Load the complement of the data on the M-bus into AC or T, as specified. Force CO to CI.

XNM K-BUS = 11

Force CO to one if the logical AND of AC and the M-bus data is non-zero. Exclusive-NOR the contents of AC with the data on the M-bus. Deposit the result in AC or T, as specified. Used to exclusive–NOR memory data with the accumulator.

F-GROUP 7 R-GROUP III

Logically OR CI with the word-wise OR of the logical AND of the contents of the specified register and the data on the I-bus and K-bus. Place the value of the carry OR on CO. Logically AND the data on the K-bus with the data on the I-bus. Exclusive-NOR the result with the contents of AC or T, as specified. Deposit the final result in the specified register.

CMA K-BUS = 00

Complement AC or T, as specified. Force CO to CI.

XNI K-BUS = 11

Force CO to one if the logical AND of the contents of AC or T, as specified, and the I-bus data is non-zero. Exclusive-NOR the contents of the specified register with the data on the I-bus. Deposit the result in AC or T, as specified. Used to exclusive-NOR input data with the accumulator.

D.C. AND OPERATING CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias . . . . 0°C to 70°C

Storage Temperature . . . . −65°C to +160°C

All Output and Supply Voltages . . . . −0.5V to +7V

All Input Voltages . . . . −1.0V to +5.5V

Output Currents . . . . 100 mA

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum ratings for extended periods may effect device reliability.

T_A = 0°C to +70°C

SYMBOL	PARAMETER	LIMITS MIN	TYP(1)	MAX	UNIT	CONDITIONS
V_C	Input Clamp Voltage (All Input Pins)		−0.8	−1.0	V	V_{CC} = 4.75V, I_C = −5 mA
I_F	Input Load Current:					
	F_0–F_6, CLK, K_0, K_1, EA, ED		−0.05	−0.25	mA	V_{CC} = 5.25V, V_F = 0.45V
	I_0, I_1, M_0, M_1, LI		−0.85	−1.5	mA	
	CI		−2.3	−4.0	mA	
I_R	Input Leakage Current:					
	F_0–F_6, CLK, K_0, K_1, EA, ED			40	μA	V_{CC} = 5.25V, V_R = 5.25V
	I_0, I_1, M_0, M_1, LI			60	μA	
	CI			180	μA	
V_{IL}	Input Low Voltage			0.8	V	V_{CC} = 5.0V
V_{IH}	Input High Voltage	2.0			V	
I_{CC}	Power Supply Current		145	190	mA	V_{CC} = 5.25V(2)
V_{OL}	Output Low Voltage (All Output Pins)		0.3	0.45	V	V_{CC} = 4.75V, I_{OL} = 10 mA
V_{OH}	Output High Voltage (All Output Pins)	2.4	3.0		V	V_{CC} = 4.75V, I_{OH} = −1 mA
I_{OS}	Short Circuit Output Current (All Output Pins)	−15	−25	−60	mA	V_{CC} = 5.0V
$I_{O(off)}$	Off State Output Current			−100	μA	V_{CC} = 5.25V, V_O = 0.45V
	A_0, A_1, D_0, D_1, CO and RO			100	μA	V_{CC} = 5.25V, V_O = 5.25V

NOTES:
(1) Typical values are for T_A = 25°C and nominal supply voltage
(2) CLK input grounded, other inputs open.

A.C. CHARACTERISTICS AND WAVEFORMS

$T_A = 0°C$ to $70°C$, $V_{CC} = 5V \pm 5\%$

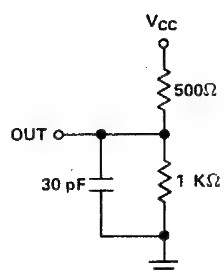
SYMBOL	PARAMETER	MIN	TYP[1]	MAX	UNIT
t_{CY}	Clock Cycle Time	100	70		ns
t_{WP}	Clock Pulse Width	33	20		ns
t_{FS}	Function Input Set-Up Time (F_0 through F_6)	60	40		ns
	Data Set-Up Time:				
t_{DS}	I_0, I_1, M_0, M_1, K_0, K_1	50	30		ns
t_{SS}	LI, CI	27	13		ns
	Data and Function Hold Time:				
t_{FH}	F_0 through F_6	5	-2		ns
t_{DH}	I_0, I_1, M_0, M_1, K_0, K_1	5	-4		ns
t_{SH}	LI, CI	15	2		ns
	Propagation Delay to X, Y, RO from:				
t_{XF}	Any Function Input		37	52	ns
t_{XD}	Any Data Input		29	42	ns
t_{XT}	Trailing Edge of CLK		40	60	ns
t_{XL}	Leading Edge of CLK	17			ns
	Propagation Delay to CO from:				
t_{CL}	Leading Edge of CLK	20			ns
t_{CT}	Trailing Edge of CLK		48	70	ns
t_{CF}	Any Function Input		43	65	ns
t_{CD}	Any Data Input		30	55	ns
t_{CC}	CI (Ripple Carry)		14	25	ns
	Propagation Delay to A_0, A_1, D_0, D_1 from:				
t_{DL}	Leading Edge of CLK		32	50	ns
t_{DE}	Enable Input ED, EA		12	25	ns

NOTE:
(1) Typical values are for $T_A = 25°C$ and nominal supply voltage.

TEST CONDITIONS:

Input pulse amplitude: 2.5 V
Input rise and fall times of 5 ns between 1 and 2 volts.
Output loading is 10 mA and 30 pF.
Speed measurements are made at 1.5 volt levels.

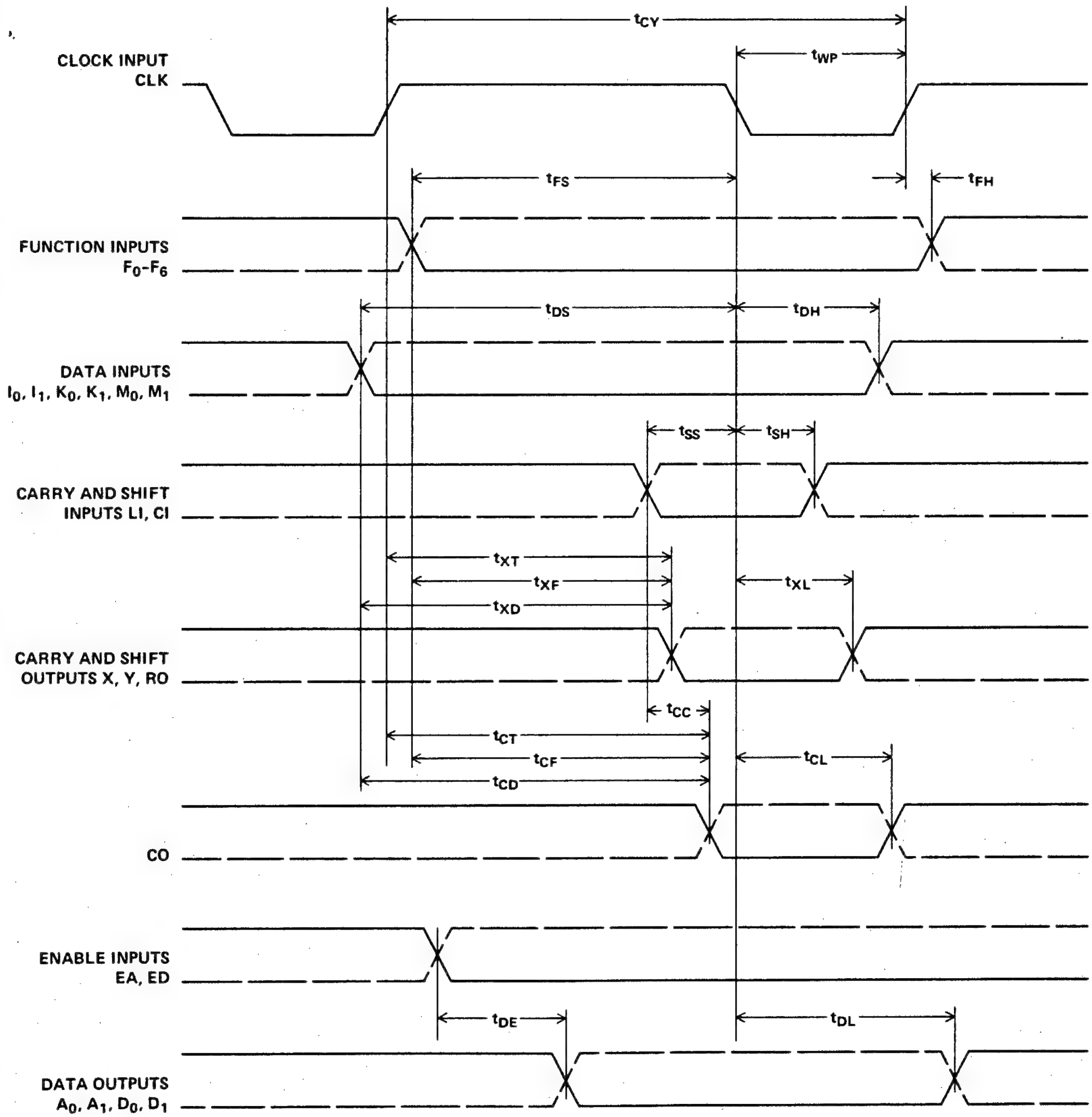
TEST LOAD CIRCUIT:



CAPACITANCE[2] $T_A = 25°C$

SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT
C_{IN}	Input Capacitance		5	10	pF
C_{OUT}	Output Capacitance		6	12	pF

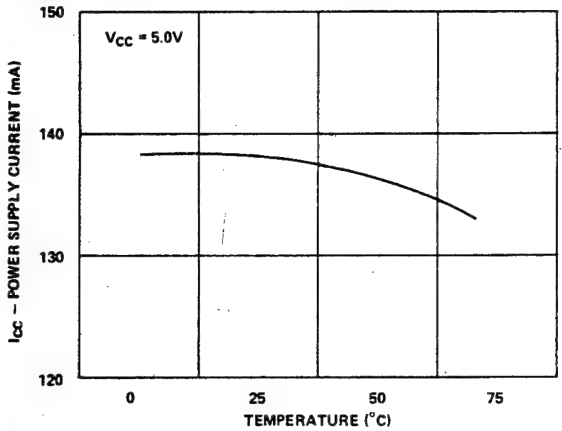
NOTE:
(2) This parameter is periodically sampled and is not 100% tested. Condition of measurement is f = 1 MHz, $V_{BIAS} = 2.5V$, $V_{CC} = 5.0V$ and $T_A = 25°C$.



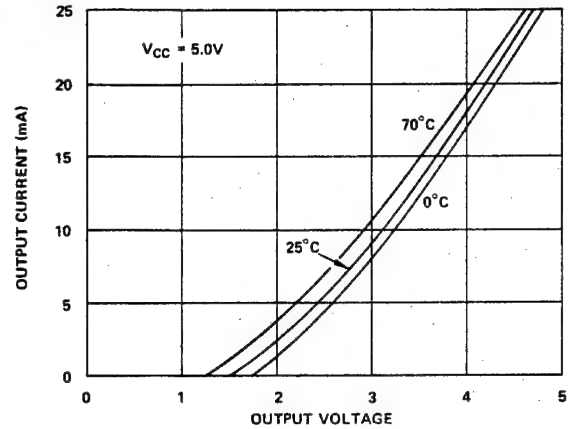
CLOCK INPUT CLK
FUNCTION INPUTS F_0-F_6
DATA INPUTS $I_0, I_1, K_0, K_1, M_0, M_1$
CARRY AND SHIFT INPUTS LI, CI
CARRY AND SHIFT OUTPUTS X, Y, RO
CO
ENABLE INPUTS EA, ED
DATA OUTPUTS A_0, A_1, D_0, D_1
t_{CY}
t_{WP}
t_{FS}
t_{FH}
t_{DS}
t_{DH}
t_{SS}
t_{SH}
t_{XT}
t_{XF}
t_{XD}
t_{XL}
t_{CC}
t_{CT}
t_{CF}
t_{CD}
t_{CL}
t_{DE}
t_{DL}

TYPICAL AC AND DC CHARACTERISTICS

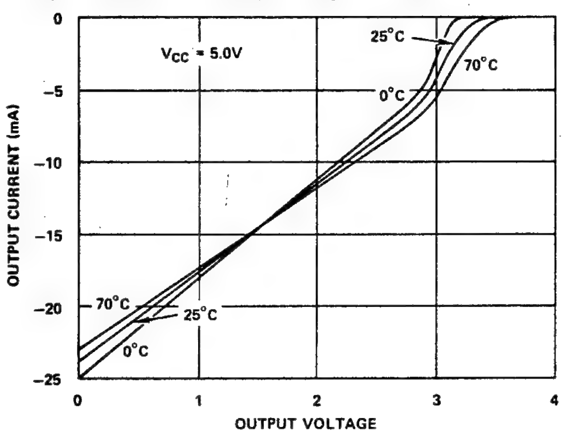
Power Supply Current vs Temperature



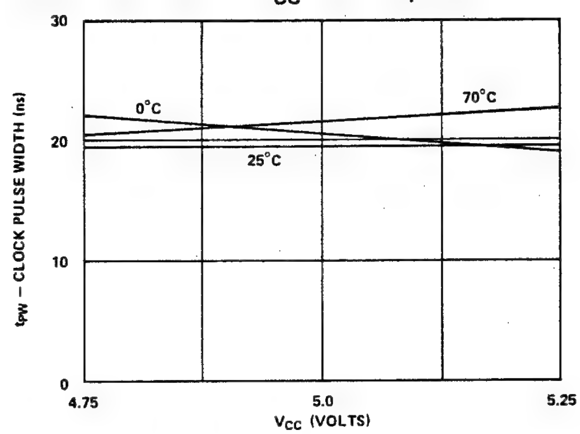
Output Current vs Output Low Voltage



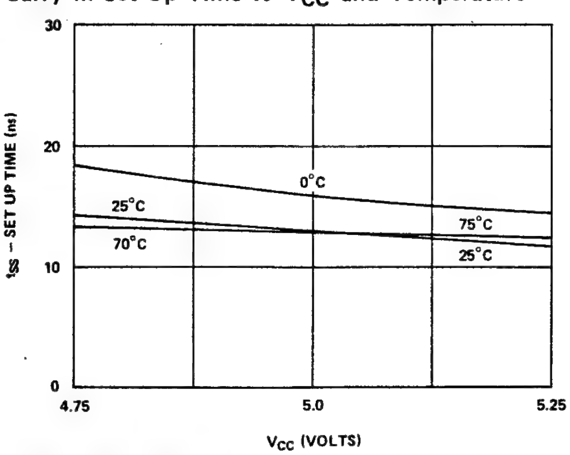
Output Current vs Output High Voltage



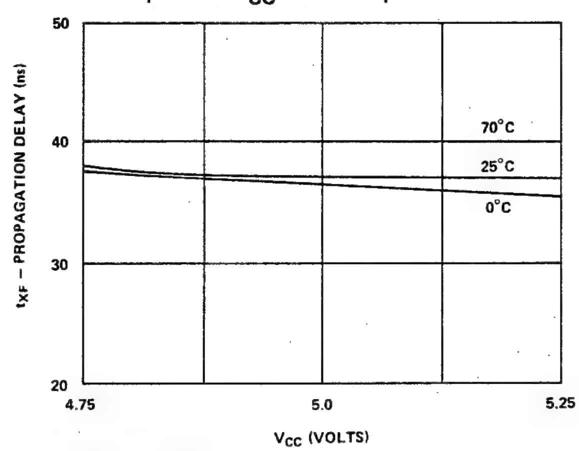
Clock Pulse Width vs V_{CC} and Temperature



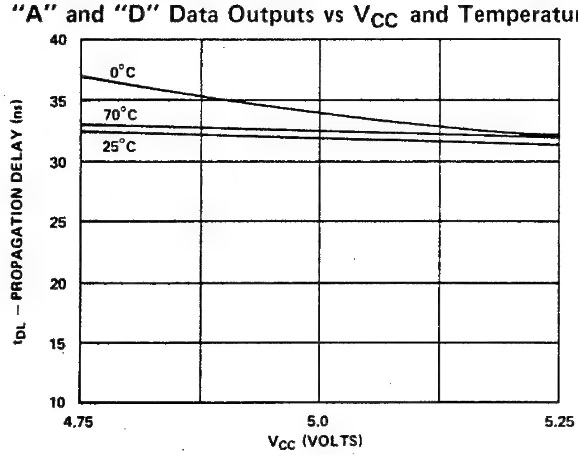
Carry in Set Up Time vs V_{CC} and Temperature



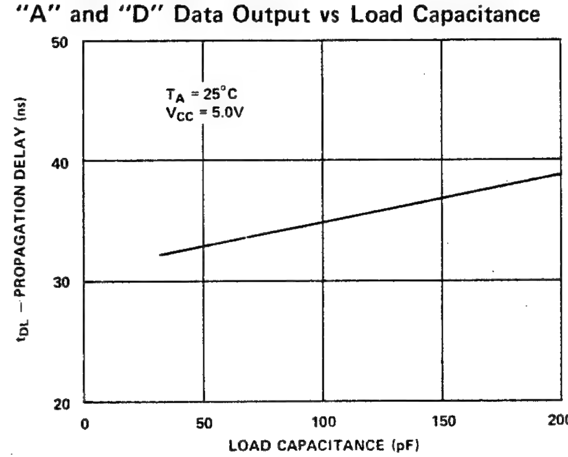
Propagation Delay Function Inputs to Cascade Outputs vs V_{CC} and Temperature



Propagation Delay Clock to "A" and "D" Data Outputs vs V_{CC} and Temperature



Propagation Delay Clock to "A" and "D" Data Output vs Load Capacitance



APPENDIX A MICRO-FUNCTION SUMMARY

F-GROUP	R-GROUP	MICRO-FUNCTION	
0	I	$R_n + (AC \wedge K) + CI \rightarrow R_n, AC$	
	II	$M + (AC \wedge K) + CI \rightarrow AT$	
	III	$AT_L \wedge (\overline{I_L \wedge K_L}) \rightarrow RO$	$LI \vee [(I_H \wedge K_H) \wedge AT_H] \rightarrow AT_H$
		$[AT_L \wedge (I_L \wedge K_L)] \vee [AT_H \vee (I_H \wedge K_H)] \rightarrow AT_L$	
1	I	$K \vee R_n \rightarrow MAR$	$R_n + K + CI \rightarrow R_n$
	II	$K \vee M \rightarrow MAR$	$M + K + CI \rightarrow AT$
	III	$(\overline{AT} \vee K) + (AT \wedge K) + CI \rightarrow AT$	
2	I	$(AC \wedge K) -1 + CI \rightarrow R_n$	(see Note 1)
	II	$(AC \wedge K) -1 + CI \rightarrow AT$	(see Note 1)
	III	$(I \wedge K) -1 + CI \rightarrow AT$	(see Note 1)
3	I	$R_n + (AC \wedge K) + CI \rightarrow R_n$	
	II	$M + (AC \wedge K) + CI \rightarrow AT$	
	III	$AT + (I \wedge K) + CI \rightarrow AT$	
4	I	$CI \vee (R_n \wedge AC \wedge K) \rightarrow CO$	$R_n \wedge (AC \wedge K) \rightarrow R_n$
	II	$CI \vee (M \wedge AC \wedge K) \rightarrow CO$	$M \wedge (AC \wedge K) \rightarrow AT$
	III	$CI \vee (AT \wedge I \wedge K) \rightarrow CO$	$AT \wedge (I \wedge K) \rightarrow AT$
5	I	$CI \vee (R_n \wedge K) \rightarrow CO$	$K \wedge R_n \rightarrow R_n$
	II	$CI \vee (M \wedge K) \rightarrow CO$	$K \wedge M \rightarrow AT$
	III	$CI \vee (AT \wedge K) \rightarrow CO$	$K \wedge AT \rightarrow AT$
6	I	$CI \vee (AC \wedge K) \rightarrow CO$	$R_n \vee (AC \wedge K) \rightarrow R_n$
	II	$CI \vee (AC \wedge K) \rightarrow CO$	$M \vee (AC \wedge K) \rightarrow AT$
	III	$CI \vee (I \wedge K) \rightarrow CO$	$AT \vee (I \wedge K) \rightarrow AT$
7	I	$CI \vee (R_n \wedge AC \wedge K) \rightarrow CO$	$R_n \overline{\oplus} (AC \wedge K) \rightarrow R_n$
	II	$CI \vee (M \wedge AC \wedge K) \rightarrow CO$	$M \overline{\oplus} (AC \wedge K) \rightarrow AT$
	III	$CI \vee (AT \wedge I \wedge K) \rightarrow CO$	$AT \overline{\oplus} (I \wedge K) \rightarrow AT$

NOTES:

1. 2's complement arithmetic adds 111 . . . 11 to perform subtraction of 000 . . . 01.
2. R_n includes T and AC as source and destination registers in R-group 1 micro-functions.
3. Standard arithmetic carry output values are generated in F-group 0, 1, 2 and 3 instructions.

SYMBOL	MEANING
I, K, M	Data on the I, K, and M busses, respectively
CI, LI	Data on the carry input and left input, respectively
CO, RO	Data on the carry output and right output, respectively
R_n	Contents of register n including T and AC (R-Group I)
AC	Contents of the accumulator
AT	Contents of AC or T, as specified
MAR	Contents of the memory address register
L, H	As subscripts, designate low and high order bit, respectively
+	2's complement addition
−	2's complement subtraction
$\wedge$	Logical AND
$\vee$	Logical OR
$\overline{\oplus}$	Exclusive-NOR
$\rightarrow$	Deposit into

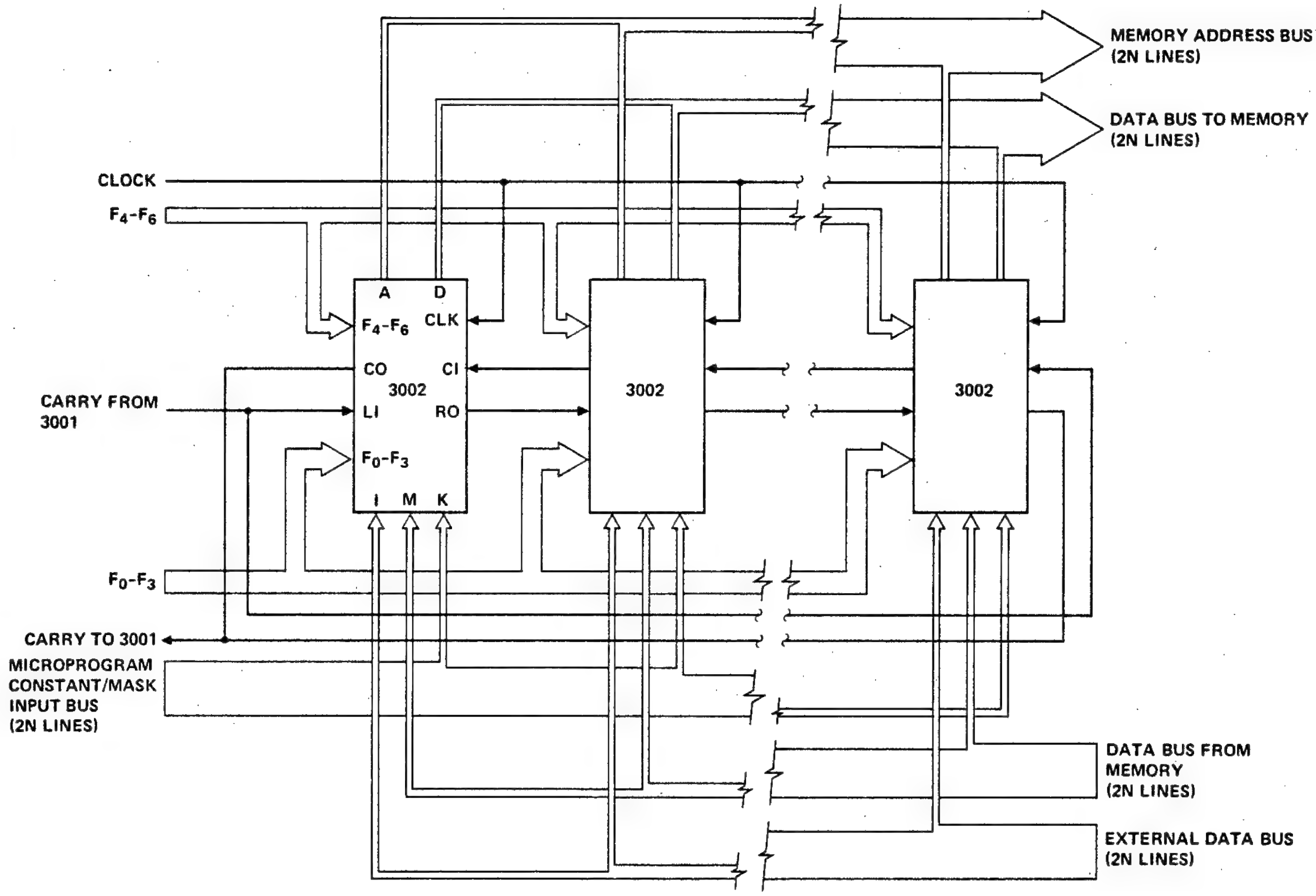
APPENDIX B ALL-ZERO AND ALL-ONE K-BUS MICRO-FUNCTIONS

K-BUS = 00 MICRO-FUNCTION		MNEMONIC	K-BUS = 11 MICRO-FUNCTION		MNEMONIC
$R_n + CI \rightarrow R_n, AC$		ILR	$AC + R_n + CI \rightarrow R_n, AC$		ALR
$M + CI \rightarrow AT$		ACM	$M + AC + CI \rightarrow AT$		AMA
$AT_L \rightarrow RO$ $\quad AT_H \rightarrow AT_L$	$LI \rightarrow AT_H$	SRA	(See Appendix B)		–
$R_n \rightarrow MAR$	$R_n + CI \rightarrow R_n$	LMI	$11 \rightarrow MAR$	$R_n - 1 + CI \rightarrow R_n$	DSM
$M \rightarrow MAR$	$M + CI \rightarrow AT$	LMM	$11 \rightarrow MAR$	$M - 1 + CI \rightarrow AT$	LDM
$\overline{AT} + CI \rightarrow AT$		CIA	$AT - 1 + CI \rightarrow AT$		DCA
$CI - 1 \rightarrow R_n$	See Note 1	CSR	$AC - 1 + CI \rightarrow R_n$	See Note 1	SDR
$CI - 1 \rightarrow AT$		CSA	$AC - 1 + CI \rightarrow AT$		SDA
(See CSA above)		–	$I - 1 + CI \rightarrow AT$		LDI
$R_n + CI \rightarrow R_n$		INR	$AC + R_n + CI \rightarrow R_n$		ADR
(See ACM above)		–	(See AMA above)		–
$AT + CI \rightarrow AT$		INA	$I + AT + CI \rightarrow AT$		AIA
$CI \rightarrow CO$	$0 \rightarrow R_n$	CLR	$CI \vee (R_n \wedge AC) \rightarrow CO$	$R_n \wedge AC \rightarrow R_n$	ANR
$CI \rightarrow CO$	$0 \rightarrow AT$	CLA	$CI \vee (M \wedge AC) \rightarrow CO$	$M \wedge AC \rightarrow AT$	ANM
(See CLA above)		–	$CI \vee (AT \wedge I) \rightarrow CO$	$AT \wedge I \rightarrow AT$	ANI
(See CLR above)		–	$CI \vee R_n \rightarrow CO$	$R_n \rightarrow R_n$	TZR
(See CLA above)		–	$CI \vee M \rightarrow CO$	$M \rightarrow AT$	LTM
(See CLA above)		–	$CI \vee AT \rightarrow CO$	$AT \rightarrow AT$	TZA
$CI \rightarrow CO$	$R_n \rightarrow R_n$	NOP	$CI \vee AC \rightarrow CO$	$R_n \vee AC \rightarrow R_n$	ORR
$CI \rightarrow CO$	$M \rightarrow AT$	LMF	$CI \vee AC \rightarrow CO$	$M \vee AC \rightarrow AT$	ORM
(See NOP above)		–	$CI \vee I \rightarrow CO$	$I \vee AT \rightarrow AT$	ORI
$CI \rightarrow CO$	$\overline{R_n} \rightarrow R_n$	CMR	$CI \vee (R_n \quad AC) \rightarrow CO$	$R_n \overline{\oplus} AC \rightarrow R_n$	XNR
$CI \rightarrow CO$	$\overline{M} \rightarrow AT$	LCM	$CI \vee (M \quad AC) \rightarrow CO$	$M \oplus AC \rightarrow AT$	XNM
$CI \rightarrow CO$	$\overline{AT} \rightarrow AT$	CMA	$CI \vee (AT \quad I) \rightarrow CO$	$I \overline{\oplus} AT \rightarrow AT$	XNI

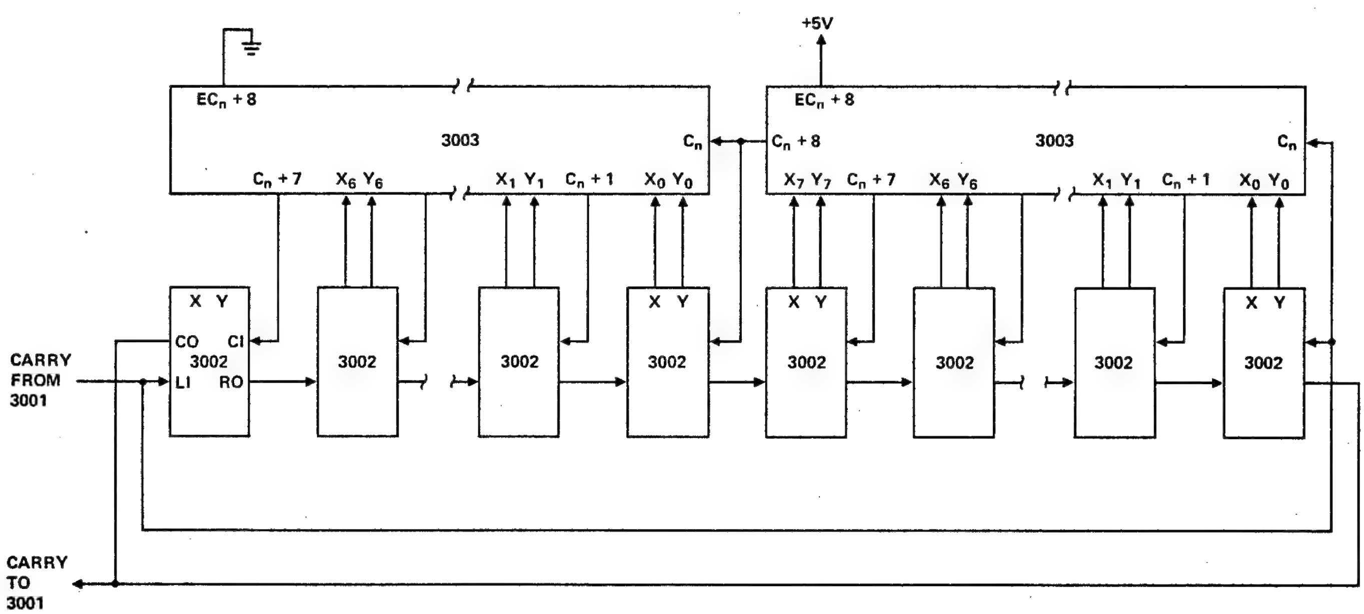
APPENDIX C FUNCTION AND REGISTER GROUP FORMATS

FUNCTION GROUP	F_6	5	4
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

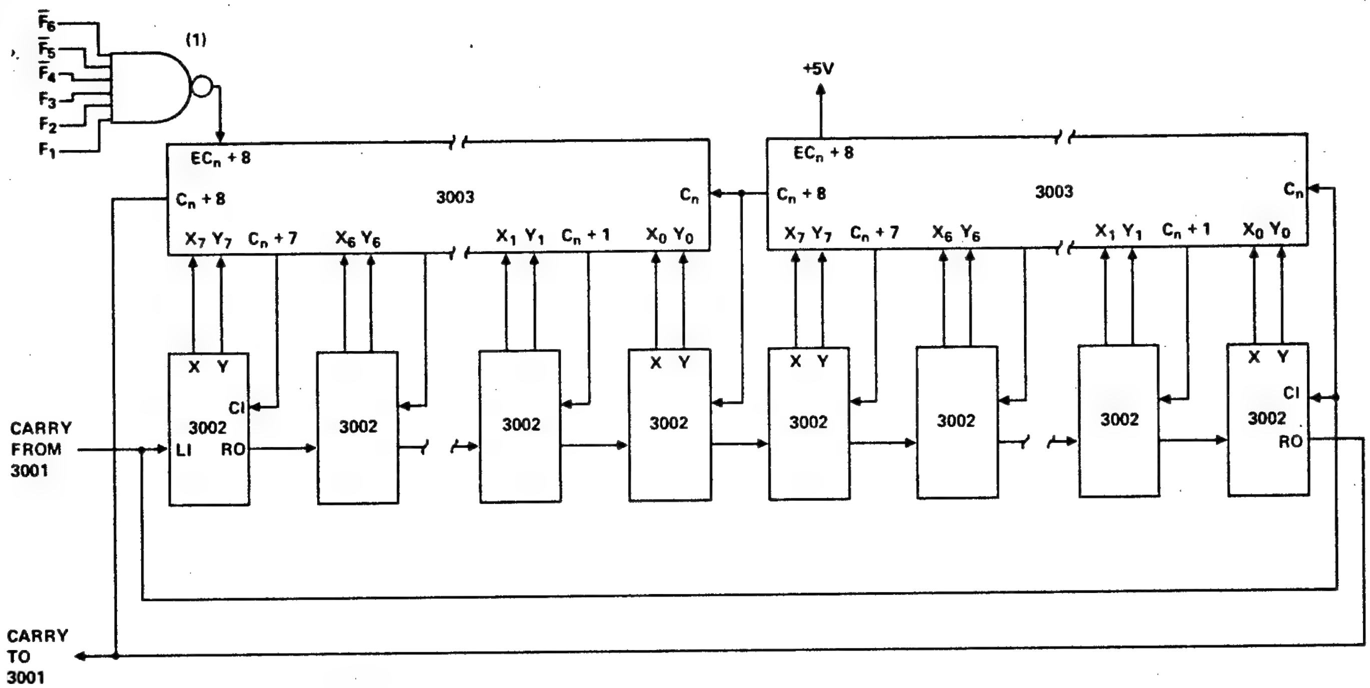
REGISTER GROUP	REGISTER	F_3	2	1	0
I	R_0	0	0	0	0
	R_1	0	0	0	1
	R_2	0	0	1	0
	R_3	0	0	1	1
	R_4	0	1	0	0
	R_5	0	1	0	1
	R_6	0	1	1	0
	R_7	0	1	1	1
	R_8	1	0	0	0
	R_9	1	0	0	1
	T	1	1	0	0
	AC	1	1	0	1
II	T	1	0	1	0
	AC	1	0	1	1
III	T	1	1	1	0
	AC	1	1	1	1



Ripple-Carry Configuration
(N 3002 CPE's)



Carry Look-Ahead Configuration
With Ripple Through the Left Slice
(32 Bit Array)



Carry Look-Ahead Configuration
With No Carry Ripple Through the Last Slice
(32 Bit Array)

NOTE:
(1) A bit from microprogram memory can be used to replace the gate shown above and inform the 3003 of an active Shift Right operation.

ORDERING INFORMATION

Part Number	Description
C3002	Central Processing Unit

intel®

Intel Corporation
3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 246-7501
TWX: 910-338-0026
Telex: 34-6372

WESTERN
1651 East 4th Street
Suite 228
Santa Ana, California 92701
Tel: (714) 835-9642
TWX: 910-595-1114

MID-AMERICA
6350 L.B.J. Freeway
Suite 178
Dallas, Texas 75240
Tel: (214) 661-8829
TWX: 910-860-5487

GREAT LAKES REGION
8312 North Main Street
Dayton, Ohio 45415
Tel: (513) 890-5350
TELEX: 288-004

EASTERN
2 Militia Drive
Suite 4
Lexington, Massachusetts 02173
Tel: (617) 861-1136
TWX: 710-321-0187

MID-ATLANTIC
520 Pennsylvania Avenue
Suite 102
Fort Washington, Pennsylvania 19034
Tel: (215) 542-9444
TWX: 510-661-3055

EUROPE
Belgium
Intel Office
216 Avenue Louise
Brussels B1050
Tel: 649-20-03
TELEX: 24814

ORIENT
Japan
Intel Japan Corporation
Kasahara Bldg.
1-6-10, Uchikanda
Chiyoda-ku
Tokyo 101
Tel: (03) 295-5441
TELEX: 781-28426

Printed in U.S.A. MCS-269-0275/27.5K

SCHOTTKY BIPOLAR LSI MICROCOMPUTER SET

3003 LOOK-AHEAD CARRY GENERATOR

The INTEL® Bipolar Microcomputer Set is a family of Schottky bipolar LSI circuits which simplify the construction of microprogrammed central processors and device controllers. These processors and controllers are truly microprogrammed in the sense that their control logic is organized around a separate read-only memory called the microprogram memory. Control signals for the various processing elements are generated by the microinstructions contained in the microprogram memory. In the implementation of a typical central processor, as shown below, the microprogram interprets a higher level of instructions called macroinstructions, similar to those found in a small computer. For device controllers, the microprograms directly implement the required control functions.

The INTEL® 3003 Look-Ahead Carry Generator (LCG) is a high speed circuit capable of anticipating a carry across a full 16-bit 3002 Central Processing Array. When used with a larger 3002 CP Array multiple 3003 carry generators provide high speed carry look-ahead capability for any word length.

The LCG accepts eight pairs of active high cascade inputs (X,Y) and an active low carry input and generates active low carries for up to eight groups of binary adders.

High Performance — 10 ns typical propagation delay

Compatible with INTEL 3001 MCU and 3002 CPE

DTL and TTL compatible

Full look-ahead across 8 adders

Low voltage diode input clamp

Expandable

28-pin DIP

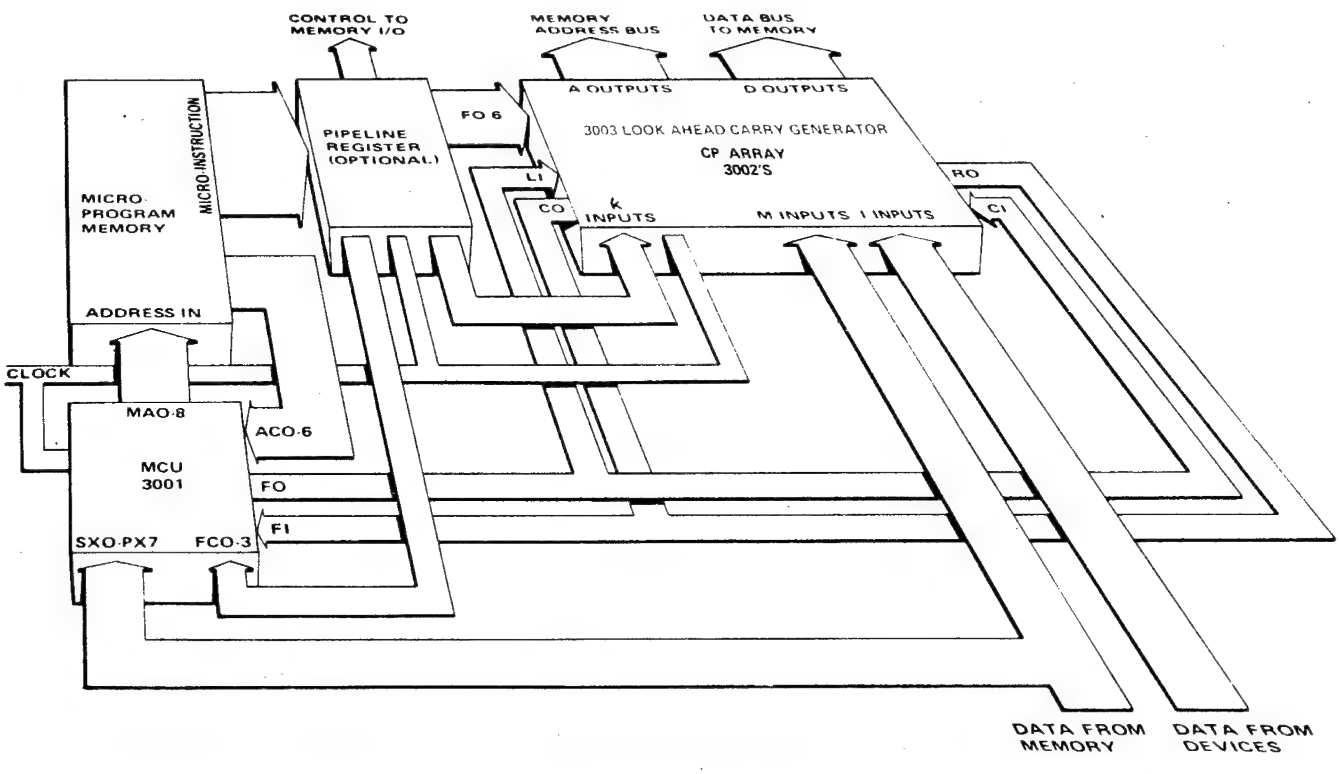


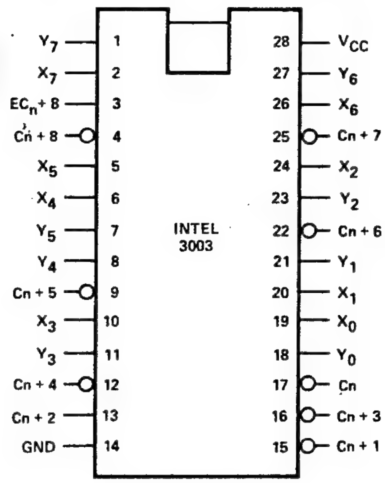
Diagram of a Typical System

Other members of the INTEL Bipolar Microcomputer Set:

3001 Microprogram Control Unit
3002 Central Processing Element
3212 Multi-Mode Latch Buffer
3214 Priority Interrupt Control Unit
3226 Inverting Bi-Directional Bus Driver
3301A Schottky Bipolar ROM (256 x 4)
3304A Schottky Bipolar ROM (512 x 8)
3601 Schottky Bipolar PROM (256 x 4)
3604 Schottky Bipolar PROM (512 x 8)

3003 LOOK-AHEAD CARRY GENERATOR

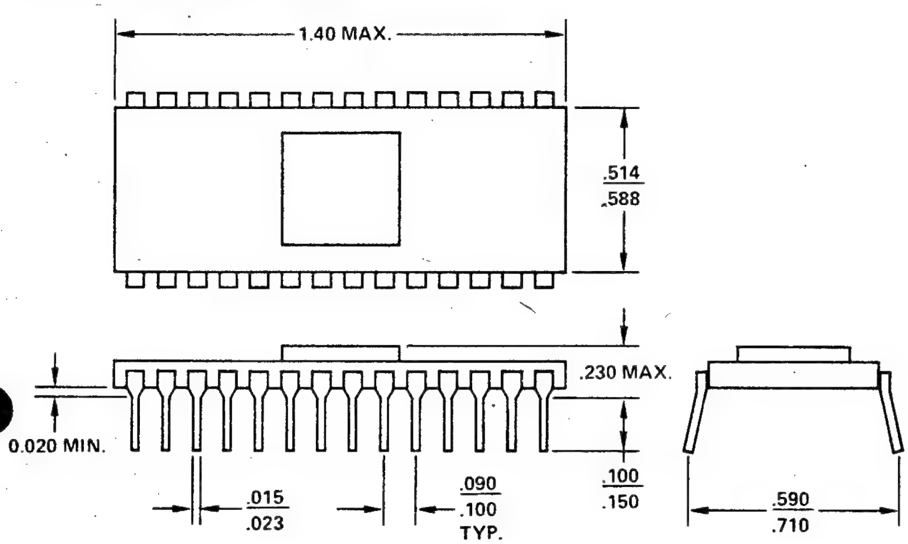
PACKAGE CONFIGURATION



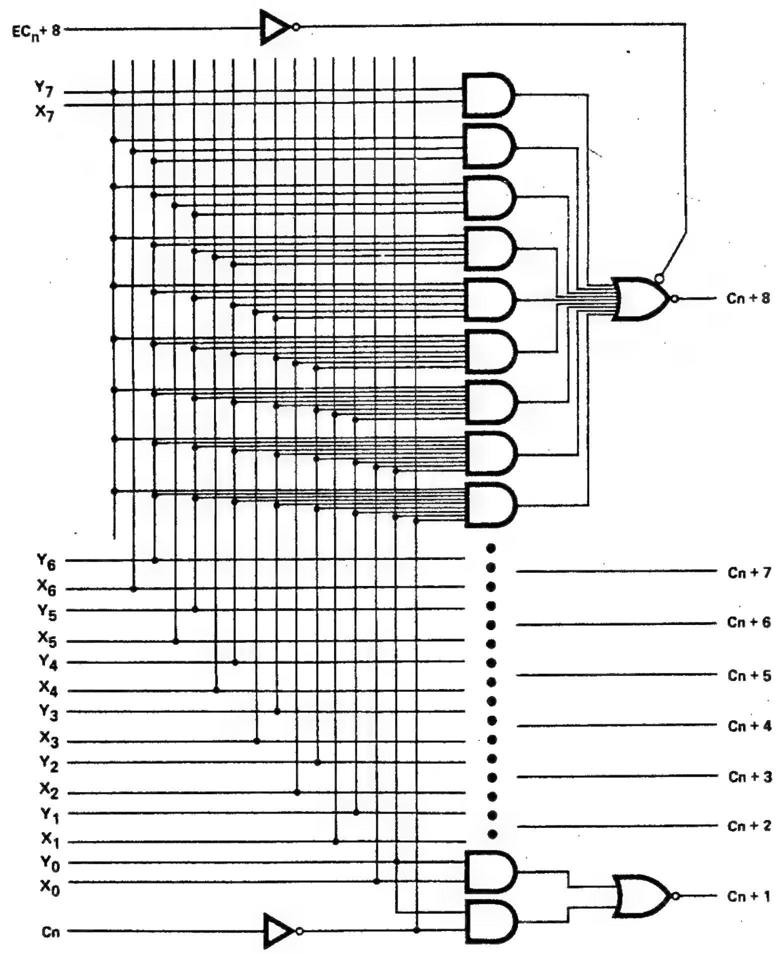
PIN DESCRIPTION

PIN	SYMBOL	NAME AND FUNCTION	TYPE
1,7,8,11,18 21,23,27	Y_0-Y_7	Standard carry look-ahead inputs	Active HIGH
2,5,6,10,19 20,24,26	X_0-X_7	Standard carry look-ahead inputs	Active HIGH
17	C_n	Carry input	Active LOW
4,9,12,13,15 16,22,25	C_{n+1}-C_{n+8}	Carry outputs	Active LOW
3	EC_{n+8}	C_{n+8} carry output enable	Active HIGH
28	V_{CC}	+5 volt supply	
14	GND	Ground	

PACKAGE OUTLINE



3003 LOGIC DIAGRAM



3003 LOGIC EQUATIONS

The 3003 Look-Ahead Generator is implemented in a compatible form for direct connection to the 3001 MCU and 3002 CPE. Logic equations for the 3003 are:

$\overline{C_n + 1} = Y_0X_0 + Y_0\overline{C}_n$

$\overline{C_n + 2} = Y_1X_1 + Y_1Y_0X_0 + Y_1Y_0\overline{C}_n$

$\overline{C_n + 3} = Y_2X_2 + Y_2Y_1X_1 + Y_2Y_1Y_0X_0 + Y_2Y_1Y_0\overline{C}_n$

$\overline{C_n + 4} = Y_3X_3 + Y_3Y_2X_2 + Y_3Y_2Y_1X_1 + Y_3Y_2Y_1Y_0X_0 + Y_3Y_2Y_1Y_0\overline{C}_n$

$\overline{C_n + 5} = Y_4X_4 + Y_4Y_3X_3 + Y_4Y_3Y_2X_2 + Y_4Y_3Y_2Y_1X_1 + Y_4Y_3Y_2Y_1Y_0X_0 + Y_4Y_3Y_2Y_1Y_0\overline{C}_n$

$\overline{C_n + 6} = Y_5X_5 + Y_5Y_4X_4 + Y_5Y_4Y_3X_3 + Y_5Y_4Y_3Y_2X_2 + Y_5Y_4Y_3Y_2Y_1X_1 + Y_5Y_4Y_3Y_2Y_1Y_0X_0 + Y_5Y_4Y_3Y_2Y_1Y_0\overline{C}_n$

$\overline{C_n + 7} = Y_6X_6 + Y_6Y_5X_5 + Y_6Y_5Y_4X_4 + Y_6Y_5Y_4Y_3X_3 + Y_6Y_5Y_4Y_3Y_2X_2 + Y_6Y_5Y_4Y_3Y_3Y_1Y_0X_0 + Y_6Y_5Y_4Y_3Y_2Y_1Y_0\overline{C}_n$

$C_n + 8$ = High Impedance State when $EC_n + 8$ Low

$C_n + 8 = Y_7X_7 + Y_7Y_6X_6 + Y_7Y_6Y_5X_5 + Y_7Y_6Y_5Y_4X_4 + Y_7Y_6Y_5Y_4Y_3X_3 + Y_7Y_6Y_5Y_4Y_3Y_2X_2 + Y_7Y_6Y_5Y_4Y_3Y_2Y_1X_1$
$+ Y_7Y_6Y_5Y_4Y_3Y_2Y_1Y_0X_0 + Y_7Y_6Y_5Y_4Y_3Y_2Y_1Y_0\overline{C}_n$ when $EC_n + 8$ high

D.C. AND OPERATING CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias . . . . 0°C to 70°C

Storage Temperature . . . . -65°C to +160°C

All Output and Supply Voltages . . . . -0.5V to +7V

All Input Voltages . . . . -1.0V to +5.5V

Output Current . . . . 100 mA

*COMMENT: Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied.

T_A = 0°C to +70°C

SYMBOL	PARAMETER	MIN.	TYP.(1)	MAX.	UNIT	CONDITIONS
V_C	Input Clamp Voltage (All Input Pins)		-0.8	-1.0	V	V_{CC} = 4.75V, I_C = -5 mA
I_F	Input Load Current: C_n and $EC_n + 8$ All Other Inputs		-0.07 -0.9	-0.25 -1.5	mA mA	V_{CC} = 5.25V, V_F = 0.45V
I_R	Input Leakage Current: C_n and $EC_n + 8$ All Other Inputs			40 100	μA μA	V_{CC} = 5.25V, V_R = 5.25V
V_{IL}	Input Low Voltage			0.8	V	V_{CC} = 5.0V
V_{IH}	Input High Voltage	2.0			V	V_{CC} = 5.0V
I_{CC}	Power Supply Current		80	130	mA	V_{CC} = 5.25V, All Y + EC_n + 8 high, All X + C_n low
V_{OL}	Output Low Voltage (All Output Pins)		0.35	0.45	V	V_C = 4.75V, I_{OL} = 4 mA
V_{OH}	Output High Voltage (All Output Pins)	2.4	3		V	V_{CC} = 4.75V, I_{OH} = -1 mA
I_{OS}	Short Circuit Output Current (All Output Pins)	-15	-40	-65	mA	V_{CC} = 5V
$I_{O(off)}$	Off-State Output Current ($C_n + 8$)			-100 +100	μA μA	V_{CC} = 5.25V, V_O = 0.45V V_{CC} = 5.25V, V_O = 5.25V

NOTE:
(1) Typical values are for T_A = 25°C and nominal supply voltage.

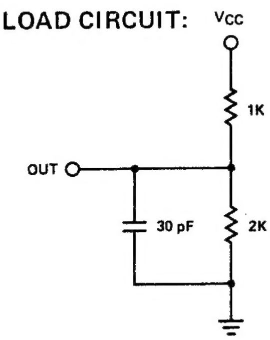
A.C. CHARACTERISTICS

$T_A = 0°C$ to $70°C$, $V_{CC} = +5V \pm 5\%$

SYMBOL	PARAMETER	MIN.	TYP.(1)	MAX.	UNIT
t_{XC}	X, Y to Outputs	3	10	20	ns
t_{CC}	Carry In to Outputs		13	30	ns
t_{EN}	Enable Time, $C_n + 8$		20	40	ns

NOTE:
(1) Typical values are for $T_A = 25°C$ and nominal supply voltage.

TEST LOAD CIRCUIT:



TEST CONDITIONS:

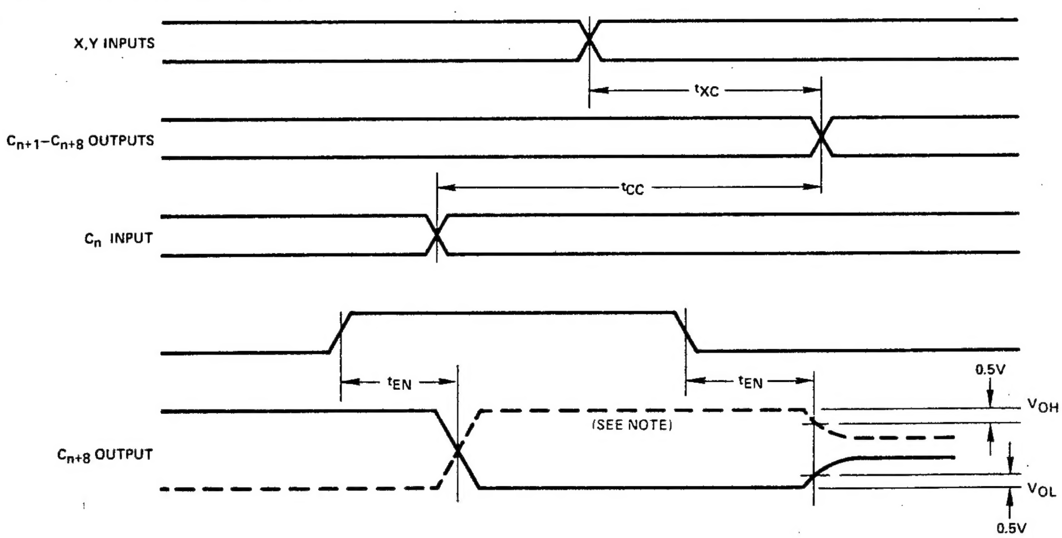
Input pulse amplitude of 2.5V.
Input rise and fall times of 5 ns between 1 and 2 volts.
Output loading is 5 mA and 30 pF.
Speed measurements are made at 1.5 volt levels.

CAPACITANCE(2) $T_A = 25°C$

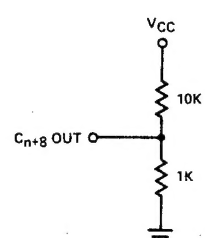
SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
C_{IN}	Input Capacitance	All inputs		12	20	pF
C_{OUT}	Output Capacitance	$C_n + 8$		7	12	pF

NOTE:
(2) This parameter is periodically sampled and is not 100% tested. Condition of measurement is $f = 1$ MHz, $V_{BIAS} = 5.0V$, $V_{CC} = 5.0V$ and $T_A = 25°C$.

3003 WAVEFORMS



NOTE: ALTERNATE TEST LOAD:



3003 TYPICAL CONFIGURATIONS

The 3003 LCG can be directly tied to the 3001 MCU and a 3002 CP array of any word length. The following figures represent typical configurations of 16- and 32-bit CP arrays. Figures 1 and 2 illustrate use of the 3003 in a system where the carry output (CO) to the 3001 MCU is rippled through the high order CPE slice. Figure 3 illustrates use of the 3003 in a system where tri-state output C_{n+8} is connected directly to the flag input on the 3001 MCU. C_{n+8} is disabled during shift right by decoding that instruction externally, thus multiplexing C_{n+8} with the shift right (RO) output of the low order CPE slice.

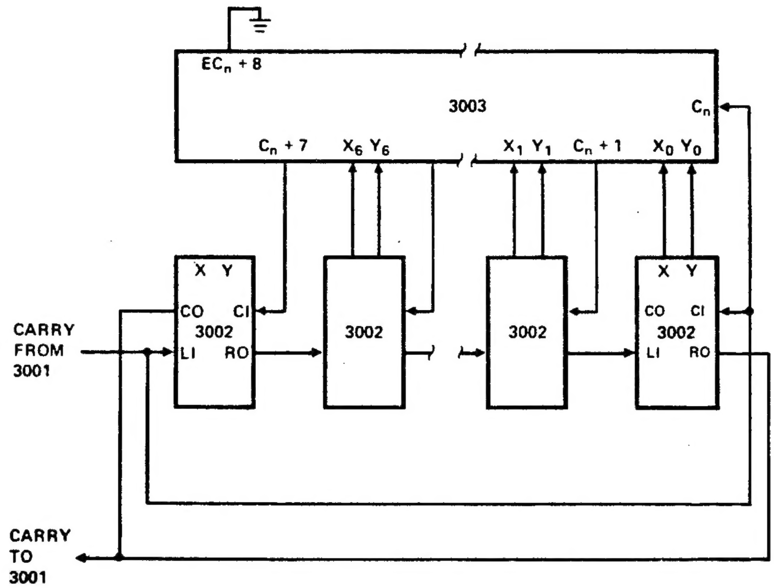


Figure 1. Carry Look-Ahead Configuration with Ripple through the Left Slice (16-Bit Array)

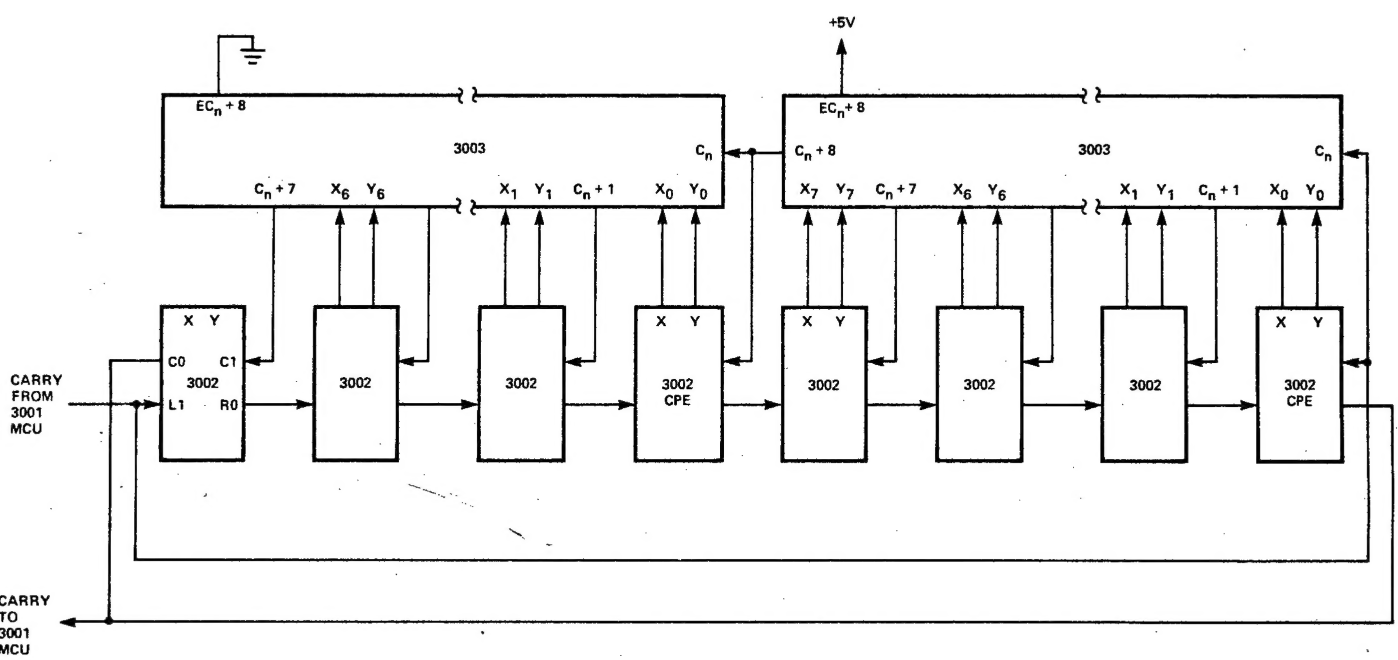
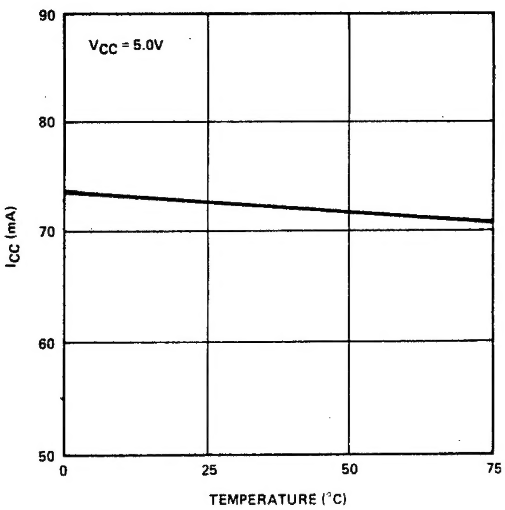
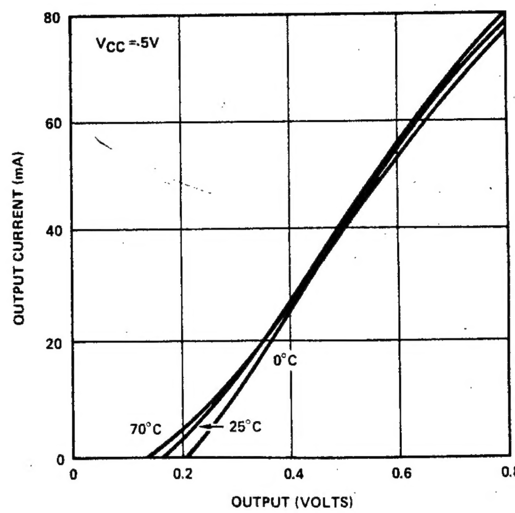


Figure 2. Carry Look-Ahead Configuration with Ripple through the Left Slice (32-Bit Array)

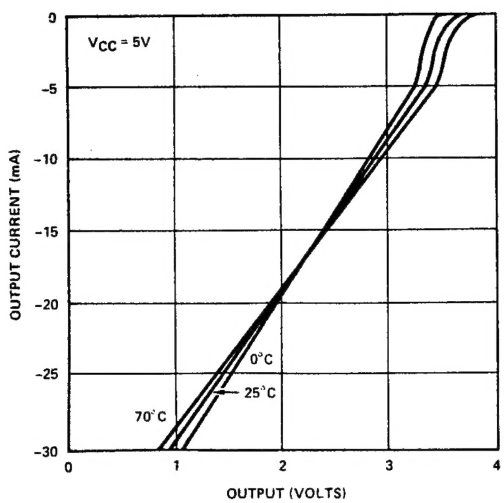
I_{CC} vs Temperature



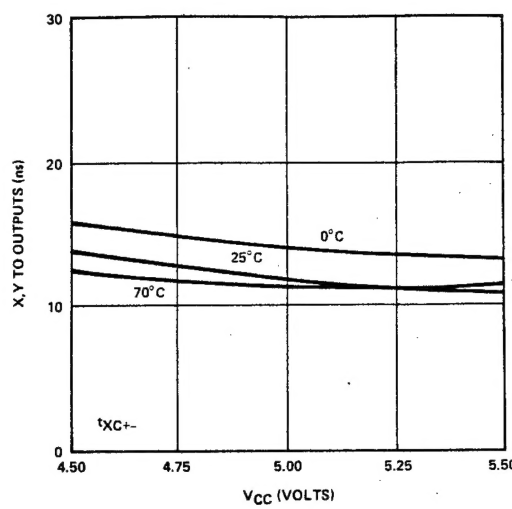
Output Current vs Output Low Voltage



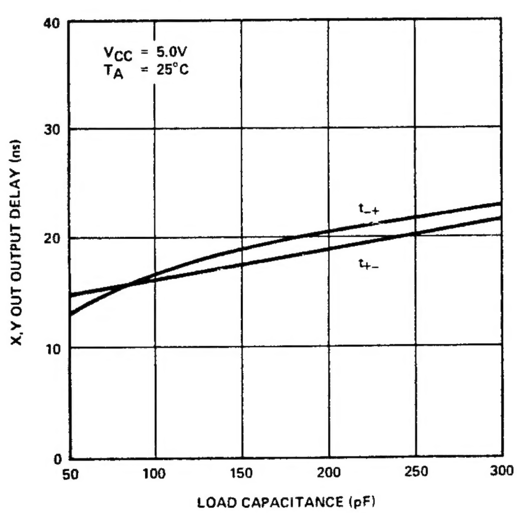
Output Current vs Output High Voltage



X,Y To Outputs vs V_{CC} & Temperature



X,Y To Output Delay vs Load Capacitance



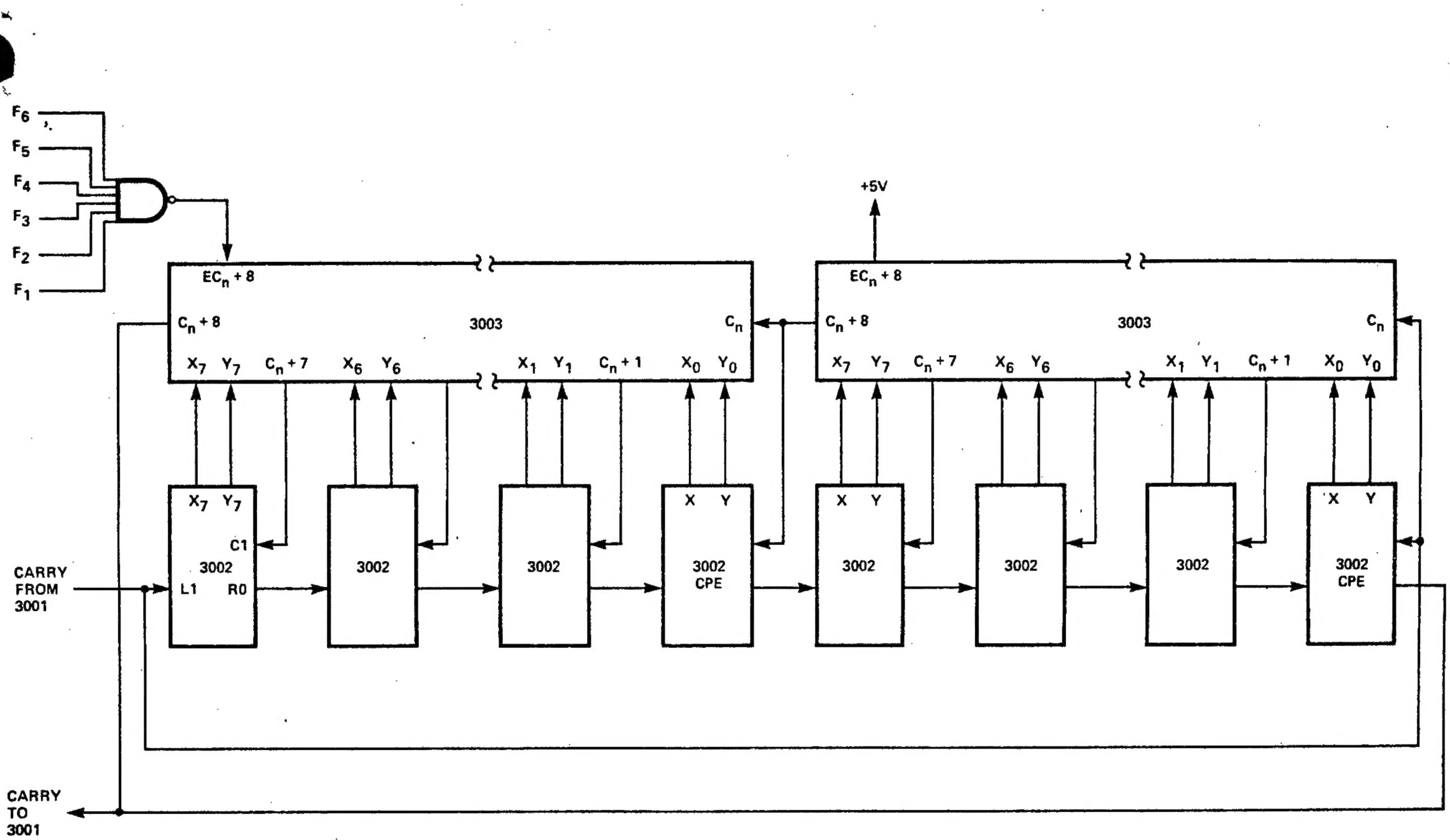


Figure 3. Carry Look-Ahead Configuration with No Carry Ripple through the Left Slice (32-Bit Array)

ORDERING INFORMATION

Part Number	Description
C3003	Look-Ahead Carry Generator

Intel Corporation
3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 246-7501
TWX: 910-338-0026
Telex: 34-6372

WESTERN
1651 East 4th Street
Suite 228
Santa Ana, California 92701
Tel: (714) 835-9642
TWX: 910-595-1114

MID-AMERICA
6350 L.B.J. Freeway
Suite 178
Dallas, Texas 75240
Tel: (214) 661-8829
TWX: 910-860-5487

GREAT LAKES REGION
8312 North Main Street
Dayton, Ohio 45415
Tel: (513) 890-5350
TELEX: 288-004

EASTERN
2 Militia Drive
Suite 4
Lexington, Massachusetts 02173
Tel: (617) 861-1136
TWX: 710-321-0187

MID-ATLANTIC
520 Pennsylvania Avenue
Suite 102
Fort Washington, Pennsylvania 19034
Tel: (215) 542-9444
TWX: 510-661-3055

EUROPE
Belgium
Intel Office
216 Avenue Louise
Brussels B1050
Tel: 649-20-03
TELEX: 24814

ORIENT
Japan
Intel Japan Corporation
Kasahara Bldg.
1-6-10, Uchikanda
Chiyoda-ku
Tokyo 101
Tel: (03) 295-5441
TELEX: 781-28426

Printed in U.S.A. MCS-349-0275/27.5K